An Automated Method For Model-Plant Mismatch Detection And Correction In Process Plants Employing Model Predictive Control (MPC)

By

Sami Saeed Ahmed Bahakim

Dissertation submitted in partial fulfillment of
the requirements for the Bachelor of Engineering (Hons)
(Chemical Engineering)

JANUARY 2012

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

An Automated Method For Model-Plant Mismatch Detection And Correction In Process Plants Employing Model Predictive Control (MPC)

By

Sami Saeed Ahmed Bahakim

Dissertation submitted in partial fulfillment of
the requirements for the Bachelor of Engineering (Hons)
(Chemical Engineering)

JANUARY 2012

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

SAMII SAEED AHMED BAHAKIM

ABSTRACT

A model-predictive controller (MPC) uses the process model to predict future outputs of the system. Hence, its performance is directly related to the quality of the model. The difference between the model and the actual plant is termed model-plant mismatch (MPM). Since MPM has significant effect on MPC performance, the model has to be corrected and updated whenever high MPM is detected. Re-identification of the process model with large number of inputs and outputs is costly due to potential production losses and high manpower efforts. Therefore, detection of the location of the mismatch is needed so that only that channel is re-identified.

Detection methods using partial correlation analysis as well as other methods have been developed, but these are qualitative methods that does not indicate the extent of the mismatch clearly and whether or not corrective action is necessary. The proposed methodology of this project uses a quantitative variable (e/u) which is the model errors divided by the manipulated variables, to identify changes in the plant gain and hence the mismatch. Taguchi experiments were carried out to identify the most contributing gains to the overall process, and then focus on these major contributors to find the threshold limits of mismatch by trial and error. When the mismatch indicated by the variable (e/u) exceeds the threshold limit, auto-correction of the model gain of the controller is made to match with the new plant gain.

The proposed method was assessed in simulations using MATLAB and Simulink on the Wood and Berry distillation column case study and was successfully validated. Testing for various mismatch scenarios for both two major contributors to the process, the algorithm was able to bring the output back to the desired set-point in a very short time.

ACKNOWLEDGEMENT

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost I offer my sincerest gratitude to my supervisor, Dr Nooryusmiza Yusoff, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I also would like to thank to the research officer and Masters student Mrs Nur Hidayah Binti Kamal for her assistance and knowledge sharing during the period that I worked in this project especially in using complex mathematical softwares.

Not forgetting thanks to Final Year Project II coordinator, Dr Lukman b Ismail and all Chemical Engineering Department lecturers for the help and beneficial professional guidance to me. Last but not least, thank you to my family for their love, support and encouragement.

Table of Contents

List of Figures

List of Tables

CHAPTER 1:
INTRODUCTION

1.1 Background study

Performance requirements for process plants have become increasingly difficult to satisfy nowadays, which in turn makes process control become increasingly important due to increased need on safe and efficient plant operation (Seborg et al., 2004). Modern process plants, designed for flexible production and to maximize recovery of energy and material, are becoming more complex. Process units are tightly coupled and the failure of a single unit can seriously degrade overall productivity. This situation presents significant control problems with the traditional regulatory PID control. The need for a more systematic approach in the implementation and integration of control systems to enhance plant operation, led to the emergence of Advanced Process Control or APC (Willis et al., 1994). Figure 1 below shows the hierarchy or process control where it shows that process control has gone to higher levels of importance and complexity with the introduction of APC in the past few decades.

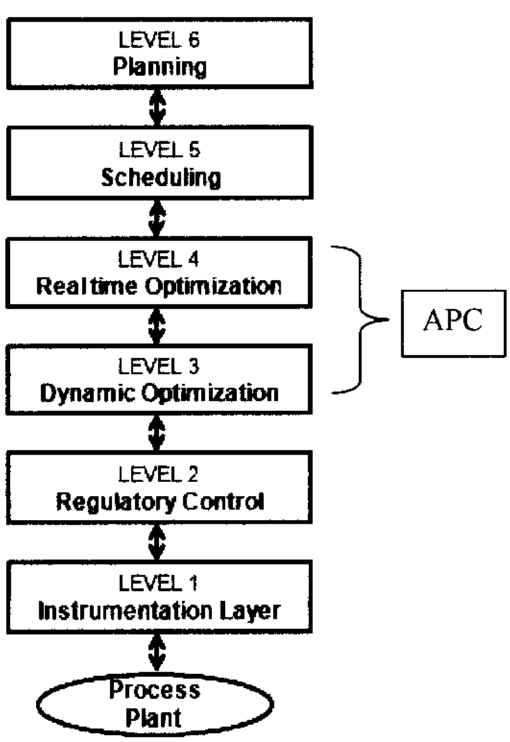


Figure 1.1: Hierarchy of process control.

APC improves product yield, quality and consistency, improve process safety and reduce environmental emissions. It has been quoted that the implementation of APC benefits 2% to 6% of operating costs (Andersen, 1992). One class of APC is model-predictive control or MPC, which uses a plant model and the current state to predict the future response of a plant. At each control interval an MPC algorithm attempts to optimize future plant behaviour by computing a sequence of future manipulated variable adjustments (Qin et al., 2003).

1.2 Model Predictive Control (MPC)

MPC has been developed and extensively used in the process industry for controlling major unit operations in chemical plants over the last two decades (Badwe et al., 2010). MPC technology has been widely and successfully applied to various processes in various industries (Kano et al., 2010). Model-predictive control (MPC), from its name suggests that the process model is being used to make the predictions on the future response and set the optimum input moves.

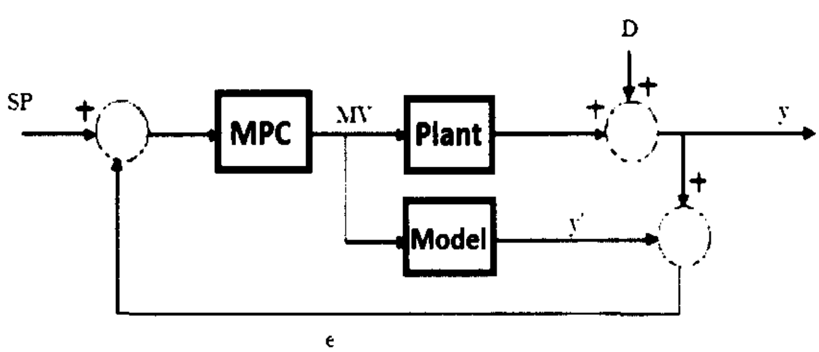


Figure 1.2: MPC control structure.

Figure 1.2 shows the MPC control structure, where two output responses **y** and **y'** are taken into account. The real plant process output is **y**, whereas **y'** is the calculated or predicted response of the plant using the model of the MPC. The difference between the two responses in the presence of disturbances is called the model residuals and is represented in the figure by the notation **e**, whose value affects the controller action.

Since the MPC model is used and affects the model residual directly which in turn determines the controller action, the closed-loop response will be dependent on the quality of the model. The model used in MPC which is usually identified at the commissioning stage of the plant is never exact with the actual plant (Badwe et al., 2009). This difference between the model and the plant is called model-plant mismatch (MPM), which degrades MPC performance and thus needs to be dealt with and corrected.

The mismatch between the process model and the plant can be detected and then corrected by carrying out re-identification to update the model. However, if the process model has large number of inputs and outputs, for example a 5 x 5 system has 120 models and to update each model becomes costly due to potential production losses as re-identification requires interfering plant tests which disturb normal operation of the plant. Therefore, it is desirable to detect the exact location of the mismatch to identify the proper channel to be updated instead of affecting the whole plant (Badwe et al., 2009).

1.3 Problem statement

From literature, there are several different proposed methods of model plant mismatch (MPM) detection, each with its own advantages and disadvantages. However, there are two problems with all proposed methods that constitute the problem statement of this research:

- A system in the proposed methods is considered mismatched according to a pre-specified 95% confidence interval. A deviation of more than 5% is considered 'mismatch' and the model needs updating when in fact the process may possibly tolerate more than that since the interval was only pre-specified and not based on studies. The actual system tolerance to mismatch maybe even non-symmetric, meaning more tolerance in one direction (positive or negative) and less in the other direction. The actual **threshold limits** to mismatch by an MPC has not yet been studied.

- The different proposed detection methods are analysed manually or qualitatively by observing graphs or charts and interpretations are based on them. No **quantitative** or automatic detection method has yet been proposed.

1.4 Objectives

- To determine the threshold limit for MPM for the process with the aid of taguchi experimental methods.
- To propose an automatic detection method for MPM using a quantitative analysis approach

1.5 Scope of the study

The study of this project will be focused mainly on improving the MPM detection method using "partial correlation analysis" proposed by Badwe et al. (2009). In addition, Taguchi experiments will be carried out to help quantify the detection of mismatch. Finally, the detected mismatch will be accounted for and corrected automatically.

The case study that will be used for this study is the Wood and Berry distillation column.

The study will be based on modeling and simulations using the software tools: MATLAB and Simulink. The focus of the study will be on the gain parameter of the model.

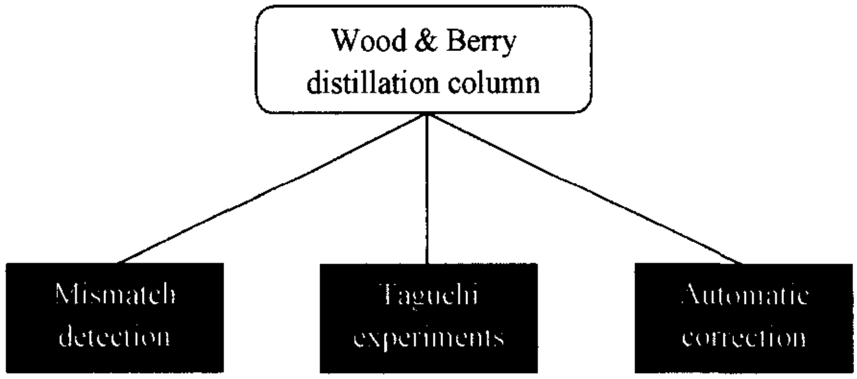


Figure 1.3: Tree diagram showing the summary of the scope of study.

CHAPTER 2:
LITERATURE REVIEW

Many have been written in literature regarding MPC and model-plant mismatch detection methods. Different detection methods proposed have different advantages over each other, and some have modified earlier methods. For the purpose of this work, this section is divided into two parts: 1) literature review on Model-predictive control (MPC) and 2) literature review on detection of model-plant mismatches.

2.1 Model-predictive control (MPC)

MPC is one type of advanced process control that widely applied in the process industries. It is classified as a method of model based controllers. Orukpe (2005) defines MPC as a form of control in which the current control action is obtained by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant.

There are a number of researchers who have reviewed on MPC history and its concepts. Morari and Lee (1996) presented an overview of MPC's early history, the current and future direction. This work proposed approximation techniques at conceptual stage for dealing with model uncertainty. The main issues among the wider research needs were identified as the areas of multivariable system identification, performance monitoring and diagnostics, non-linear state estimation and batch system control.

Qin and Badgwell (2003) have done a survey of industrial MPC technology. The survey includes the brief history of MPC, the survey of MPC control and identification technology products, applications of MPC as well as the limitations of the existing technology.

2.2 Detection of model-plant mismatch

Badwe et al. (2009) proposed a method for detecting the model-plant mismatch using partial correlation analysis between the manipulated variable and the model residuals.

$$e = \Delta u + D$$

From the above equation, where e represents the model residuals, u is the manipulated variable, Δ is the model plant mismatch and D is the disturbance, it can be seen that a correlation between the model residuals and manipulated variable indicates the extent of Δ or model plant mismatch. This is the basis principle for detecting mismatch in this method.

The proposed method was demonstrated against three case studies; a 3x3 problem, Shell control problem and an industrial case study of Kerosene Hydrofiner Unit (KHU), for gain, time delay and also no mismatch (to test for false detection). The result of the correlation plot is observed where longer lines indicate higher correlation which in turn indicates higher MPM as shown in the figure below.

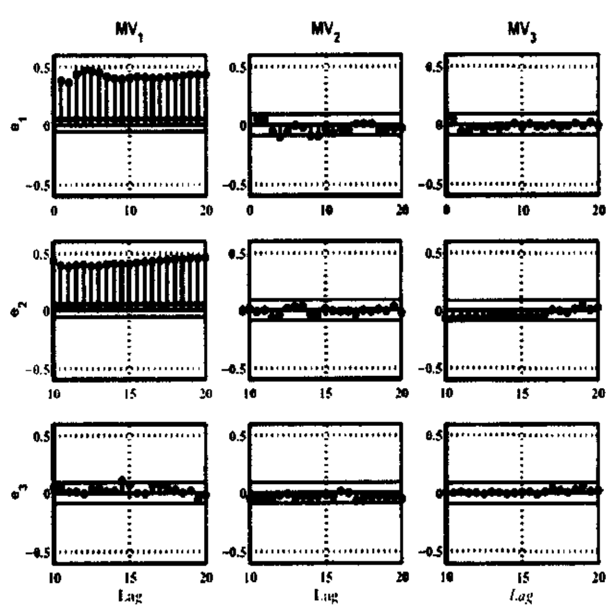


Figure 2.1: Partial correlation plots for a 3 x 3 system (*adapted from* Badwe et al.).

The two blue lines in the figure indicate the limit by which MPM is allowed by the system, which is pre-specified to allow 5% deviation only. The method proved to detect the location of model plant mismatch effectively when using partial correlation but not exactly when regular correlation analysis was employed because of the possible correlations between the manipulated variables itself. One of its advantages is that it uses routine plant operating data without the need for disturbing the plant operation.

Another detection method proposed, which was obtained from a Japanese literature where Kano et al. (2010) proposes a novel method for the detection of significant model plant mismatch using routine plant data. The proposed method uses the stepwise method to select past inputs that contribute greatly to each model residual. Large number of inputs selected means presence of model mismatch in that sub-model. The method was tested on the Wood and Berry case study where R - x_D and S-x_D models are manually mismatched and the result shown in the figure below.

Table 2.1: Table showing the result of the stepwise detection method (*adapted from* Kano et al.).

output	x_D			x_B		
input	R	S	F	R	S	F
variance	the number of selected variables					
0	4	16	1	14	9	11
0.025	4	9	1	0	0	0
0.50	2	2	0	0	0	0
0.80	1	2	0	0	0	0
2.50	1	1	0	0	0	0
score	3.48	4.18	0.18	0.00	0.00	0.00
method	the norm of coefficients					
MRA	0.85	0.34	0.61	0.19	0.07	0.33
PLS	0.69	0.37	0.26	0.10	0.03	0.14
CA	2.53	2.55	2.72	2.39	1.01	6.10
impulse	1.05	1.07	0.02	0.02	0.02	0.02

The score given to each model is proportional to the number of inputs selected, or in other words to the extent of mismatch. From Figure 2.2, the proposed method was not only able to exact location of mismatch but also gave better results when compared to other methods such as MRA (Multiple Regression Analysis), PLS (Partial Least Square) and CA (Correlation Analysis), on two case studies for gain and time constant (delay) mismatches.

Another different approach for model plant mismatch detection was proposed by Selvanathan and Tangrila (2010) by introducing a new parameter called the Plant Model Ratio (PMR) which is the ratio of the plant transfer function over the model transfer function in the frequency domain, as a measure for model-plant mismatch. The reason for using the PMR measure instead of the conventional mismatch measure (difference between plant and model outputs) is because it suits the proposed method's aim of not only identifying a mismatch, but also identifying which parameter in the model is mismatched; gain, time constant or time delay. A minimal change in set point is needed to estimate the PMR and by mapping and observing certain signatures the mismatched parameter or combination of parameters can be identified since each combination have its own specific signature form. Examples of the signatures are shown in Figure 2.3.

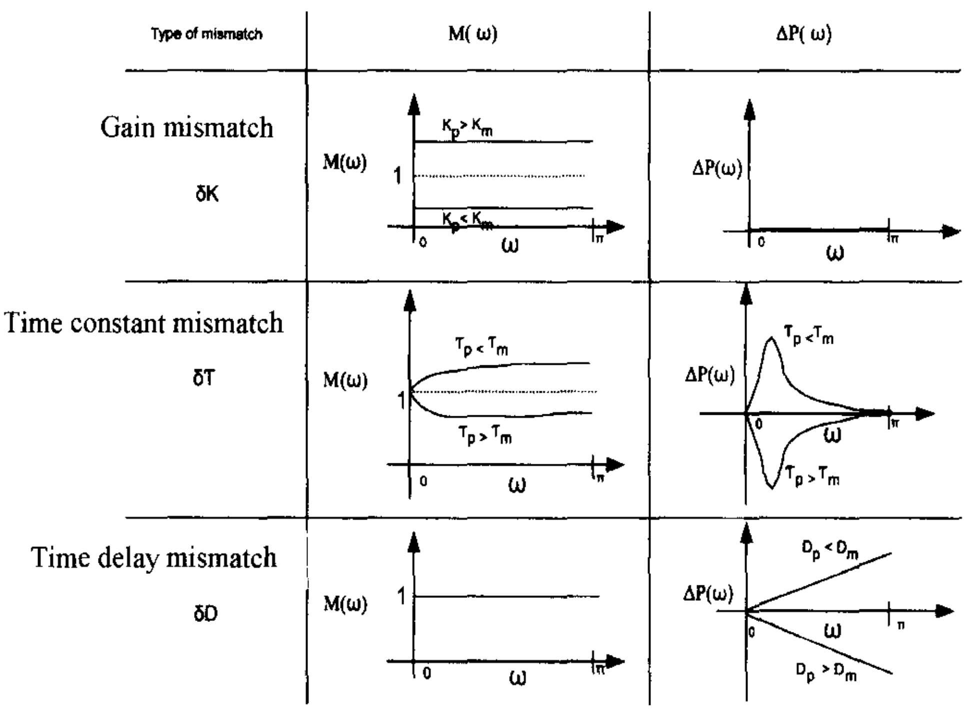


Figure 2.3: Figure showing the unique signatures for each type of mismatch (*adapted from* Selvanathan et al.).

The method was demonstrated on SISO systems, but it can be further applied on MIMO systems by the same method. The advantage of this method overs is that it specifies exactly what are the parameters that are mismatch so that it can be appreciated during updating of the models.

Others such as the work of Wang and Wang (2010) proposed a method for detecting the presence of model plant mismatch in a Dynamic Matrix Controller (DMC) system for the Wood and Berry distillation column case. The proposed methodology uses the relationship between the output error and the disturbance, followed by statistical inference to detect the model plant mismatch.

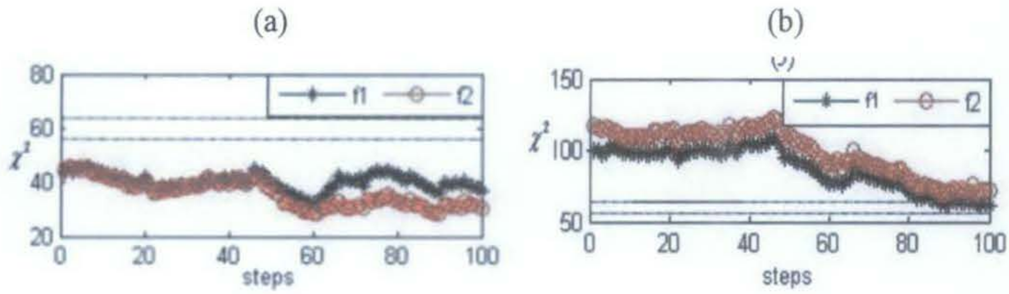
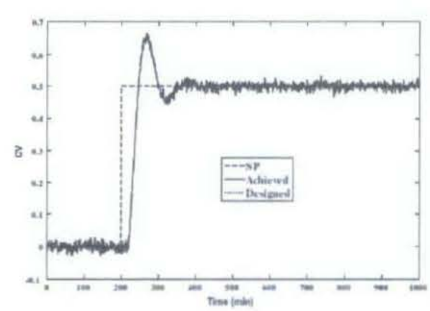


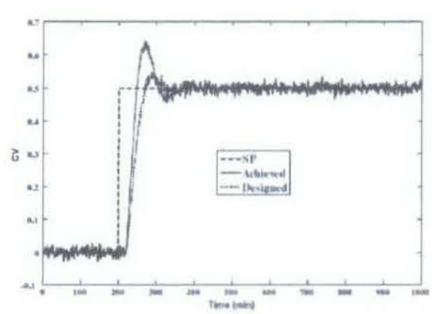
Figure 2.4: Curves obtained form complicated statistical method (a) no mismatch (b) mismatched (*adapted from* Wang and Wang).

Badwe et al. (2010) proposed a method for quantifying the significance of the impact of MPM to MPC performance degradation with regards to other causes such as improper controller tuning or disturbances. The method suggested analyses three closed-loop relationships, and based on certain diagnosis rules, the degradation of the controller can be identified where it is contributed from significantly.

(a) No mismatch and properly tuned.



(b) Have MPM



(c) Have high MPM and poorly tuned.

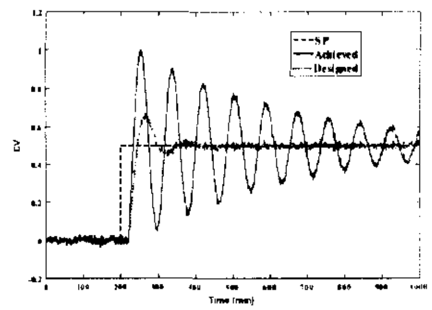


Figure 2.5 (a), (b) and (c): Responses showing the extent of the effect of MPM on MPC performance (*adapted from* Badwe et al.).

Also in the study was the effect of the direction of the perturbations (set points change). The advantage of the proposed methodology is that is the ability to estimate the designed control error from closed-loop data so that effect of MPM on controller with disturbances can be compared with the effect of deign controller with disturbances only (no MPM).

Webber and Gupta (2008) proposes a method for MIMO systems to identify which subsets of models are required to undergo re-identification due to model mismatch. The method suggested involves finding the cross-correlation between the set-point and prediction error, where the presence of mismatch is indicated by the significance of the correlations. Two examples for the Shell heavy oil fractionators case study were used to demonstrate the effectiveness of the proposed method. For a single model mismatch, it was easily identified (example 1), however, when three models were mismatched, four models were screened out as the ones that could possibly involve mismatch and had to go for further screening methods to identify the actual mismatched models. The advantage of the proposed method is that it does not involve complicated parametric calculations.

Finally Nafsun & Yusoff (2011) demonstrated the effects of model-plant mismatch on MPC performance for the Wood and Berry distillation column case. Mismatches were introduced to the gain, reverse gain, time constant and time delays and the resulting control response for set-point tracking was observed. The findings showed that gain mismatch (positive or negative) had significant effects on the controller performance as compared to time constant or time delay mismatches.

All these different methods of identifying and detecting the model-plant mismatch suggest the importance of this work and the need to find better and more efficient ways in the future.

CHAPTER 3:
METHODOLOGY

This chapter discusses the methods or ways of achieving the desired objectives. It describes the different project works involved, the milestones and how are they going to be achieved step by step.

3.1 Research methodology

The figure below shows the flowchart of the key milestone work that will be followed in carrying out this project.

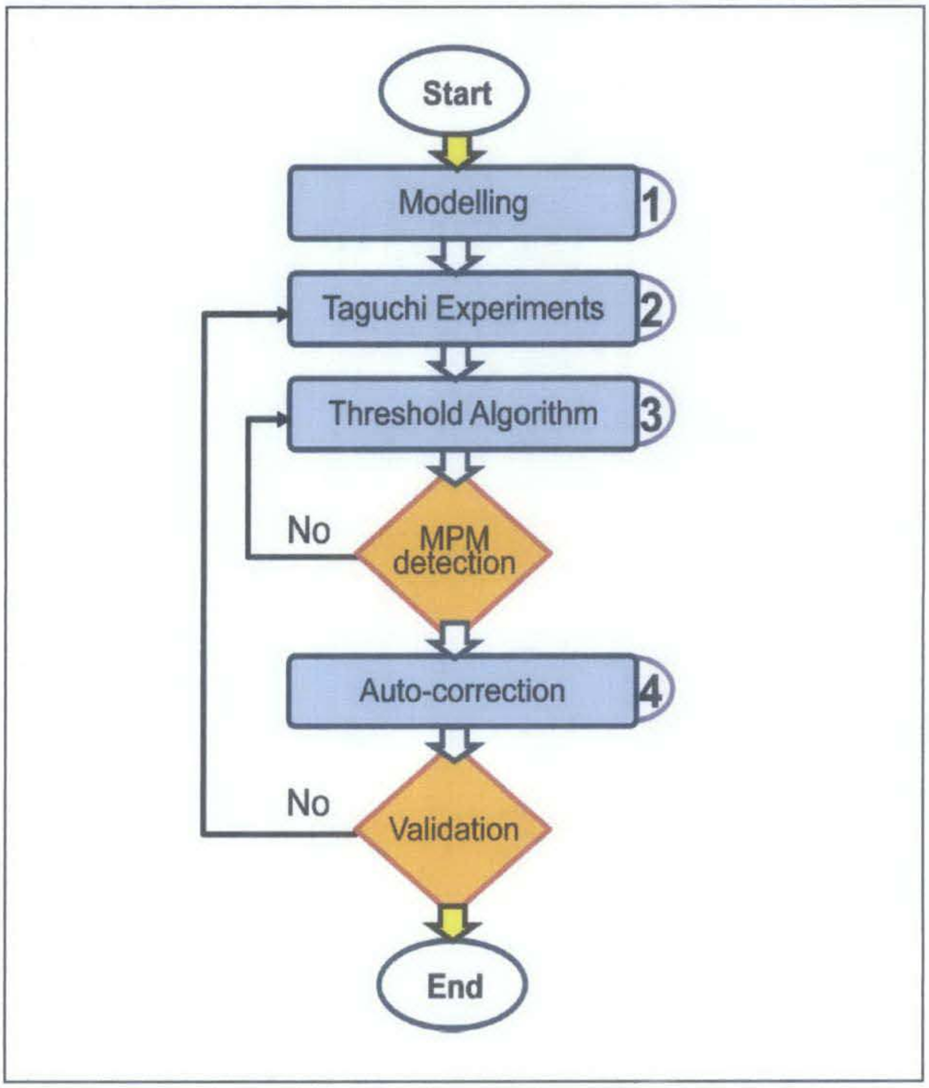


Figure 3.1: Flowchart of step by step research methodology.

3.1.1 Modelling and simulation

Modelling for the case studies and simulating it in MATLAB and SimuLink is needed at the beginning. The simulation will be done for the partial correlation analysis method first where the coding will be done in MATLAB and connected to SimuLink. The simulation to be done for the model predictive controller with the obtained models from tests or literature entered in the controller simulation. From the simulation, we can monitor the output whether it is within acceptable limits.

3.1.2 Taguchi experiments

The purpose of the taguchi experiments is to conduct several different experiments to test different combinations of factors and levels (for our case gains and gain values), to come up with TWO findings:

i. Which of the different gains in the system contribute most to the outputs of the process.
ii. Which combinations of gain values and mismatches are acceptable and which are not.

3.1.3 Threshold algorithm

This is an important milestone in the whole project. After having succeeded in all of the above steps using the pre-specified 95% confidence interval, now the allowable mismatch range is increased incrementally and the response is observed. The tolerance of the process to the increased mismatch is observed and assessed whether it is considered satisfactory or not at each increment. By this method, a point will be reached where the increased increment in mismatch caused a large deviation in the response, defining the system as unstable or unsatisfactory. This is the point whereby the threshold limit will be set. From this trial and error experiments together with the taguchi results, we can go further and develop threshold algorithm involving different combinations of gain mismatch instead of threshold limits of individual gains.

3.1.4 MPM detection

Next is to determine whether our simulation can in fact detect a model plant mismatch if present. At the start the testing is carried out by following the literatures in setting a 95% confidence interval above which is considered a mismatch. To check our simulation is correct and detects a mismatch, the model parameters are intentionally altered in the simulation (for gain, time constant and time delay) and check the system for set point tracking as well as disturbance rejections. Unless and until the simulation succeeds in detecting the MPM using the code based on the desired method, then only the next step is carried out.

3.1.5 Auto-correction

After having obtained the threshold limit/algorithm for the process, the next final step in the modification is to develop an algorithm that can identify when the limit has been surpassed automatically without having to observe at response curves or any other. This coding will most probably be done in MATLAB and connected to Simulink. The main principle behind auto-correction is to identify model plant changes continuously and correcting it as such, using some measurable indicator value that tells us where our gain is at the moment.

3.1.6 Validation

Finally, validation of the established program must be validated on the case study named earlier. Once validated, the method is considered working and can be ready for publication.

Tools required: MATLAB and SimuLink.

CHAPTER 4:
RESULTS & DISCUSSION

4.1 Project work

Before displaying the actual results obtained, the exact work done in this project is summarized below:

- Obtained the model of the process for the case study.
- Simulated the Wood and Berry distillation column in Simulink.
- Conducted the Taguchi experiments to determine the significant contributors to changes in the process output.
- Carried out simulations to determine the threshold limit for each process gain.
- Developed an algorithm for the detection of MPM for the simulation produced earlier based on threshold limits.
- Obtained an indicator variable based on some relationships that are measurable and can be related directly to the plant gain.
- Developed an algorithm to detect mismatch and correct it automatically when the threshold limit is exceeded.

4.2 Modelling

We need TWO process models, one for each of our case studies which are the Wood and Berry distillation column, and Shell heavy oil fractionators. The models would be needed for carrying out simulations later in our study. Fortunately, there is sufficient data in literature about the models of both processes such as that found in Badwe et al. (2010) as well as Kano et al. (2010).

Wood and Berry

$$\begin{bmatrix} X_D(s) \\ X_B(s) \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} R(s) \\ S(s) \end{bmatrix} + \begin{bmatrix} \dfrac{3.8e^{-8.1s}}{14.9s+1} \\ \dfrac{4.9e^{-3.4s}}{13.2s+1} \end{bmatrix} F(s)$$

For the time being studies have been carried out for the Wood and Berry model only, so from here onwards, all process models and studies will be referred to this case study.

4.3 Simulation of the process using MATLAB and Simulink

The process is simulated for the 2 x 2 Wood and Berry distillation column using Simulink block diagrams to represent the different process transfer functions as displayed in the figure below.

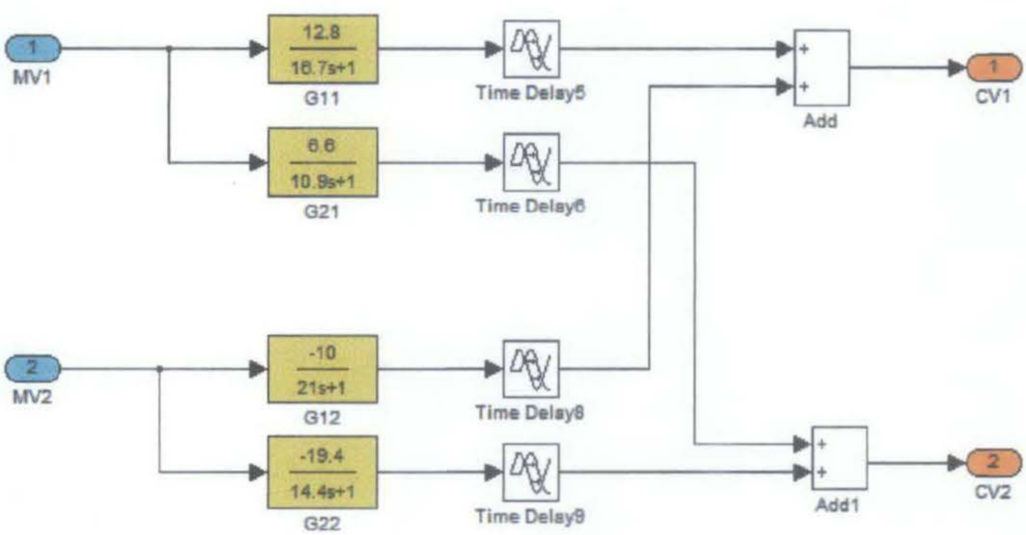


Figure 4.1: SimuLink block diagram representing the process.

As can be clearly seen from the figure, there are 2 manipulated variables (MVs) controlling 2 controlled variables (CVs). Time delays for each process have been added separately. Since in this project the usage of a model predictive controller (MPC) is desired, the above structure we will be repeated twice; one representing the actual plant, and the other representing the model we established.

By using MATLAB/Simulink, the models with MPC controller are built based on the closed-loop internal model control (IMC) structure. In designing MPC controller, two items need to be specified: 1) tuning parameters and 2) a discrete process model to represent the actual plant. The values for tuning parameters are based on the previous work by Badwe et al. (2009) and Seborg et al. (2004) but these parameters are different depends on the mismatch scenarios. The tuning parameter of MPC controller is supplied to the controller by linking Simulink model with MATLAB script. The tuning parameters for all case studies are shown in Table 4.1.

Table 4.1: Table showing the tuning parameters for the MPC (*adapted from Nafsun & Yusoff*).

Tuning Parameters	Case Study 1
Control horizon, M	2
Prediction horizon, P	30
Sampling period, Δt	1
Constraints:	
-Input	[ ]
-Input rate	[-10 10;-10 10]
-Output	[-20 20;-20 20]
Weighting matrices:	
-Input	diag[0 0]
-Input rate	diag[0.1 0.1]
-Output	diag[1 10]

The parameters in the above table is implemented and inserted into the MPC toolbox of Simulink via MATLAB coding shown in APPENDIX. The whole simulation block of the Wood and Berry process as done in Simulink is shown in Figure 4.2.

The simulation shown here is that of Set-Point Tracking, where we change the set point of the system while assuming no disturbance is affecting the system. The blue blocks in the figure represents the process and model of the system that was shown earlier in Figure 4.1. A step input of 3 is implemented on the system and the output response is observed through the Scope block as shown in Figure 4.3.

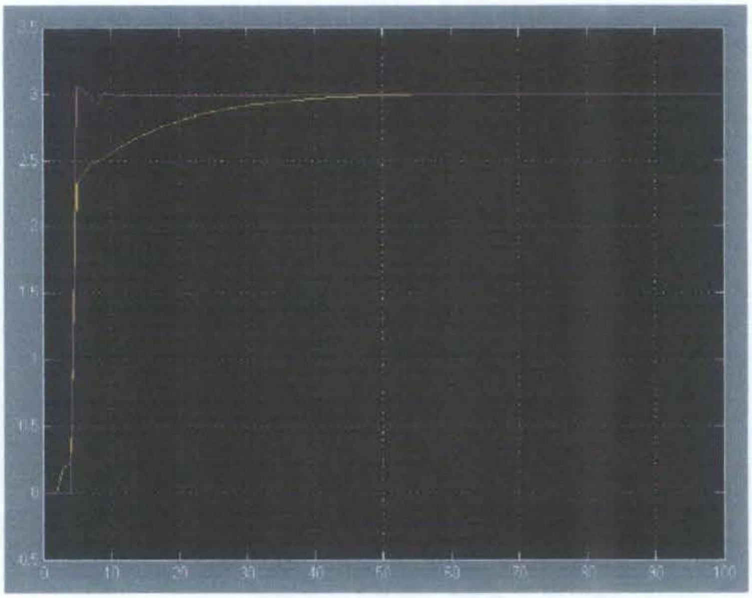


Figure 4.3: Scope showing both output responses for a step input of 3.

The process is simulated for 100 time units while the MPC tries to bring the outputs (CV1 and CV2) to the new set point. For carrying out different calculations in our study later, data are collected from the simulation via the green boxes shown in Figure 4.2 that sends the simulation result as an array to the MATLAB Workspace.

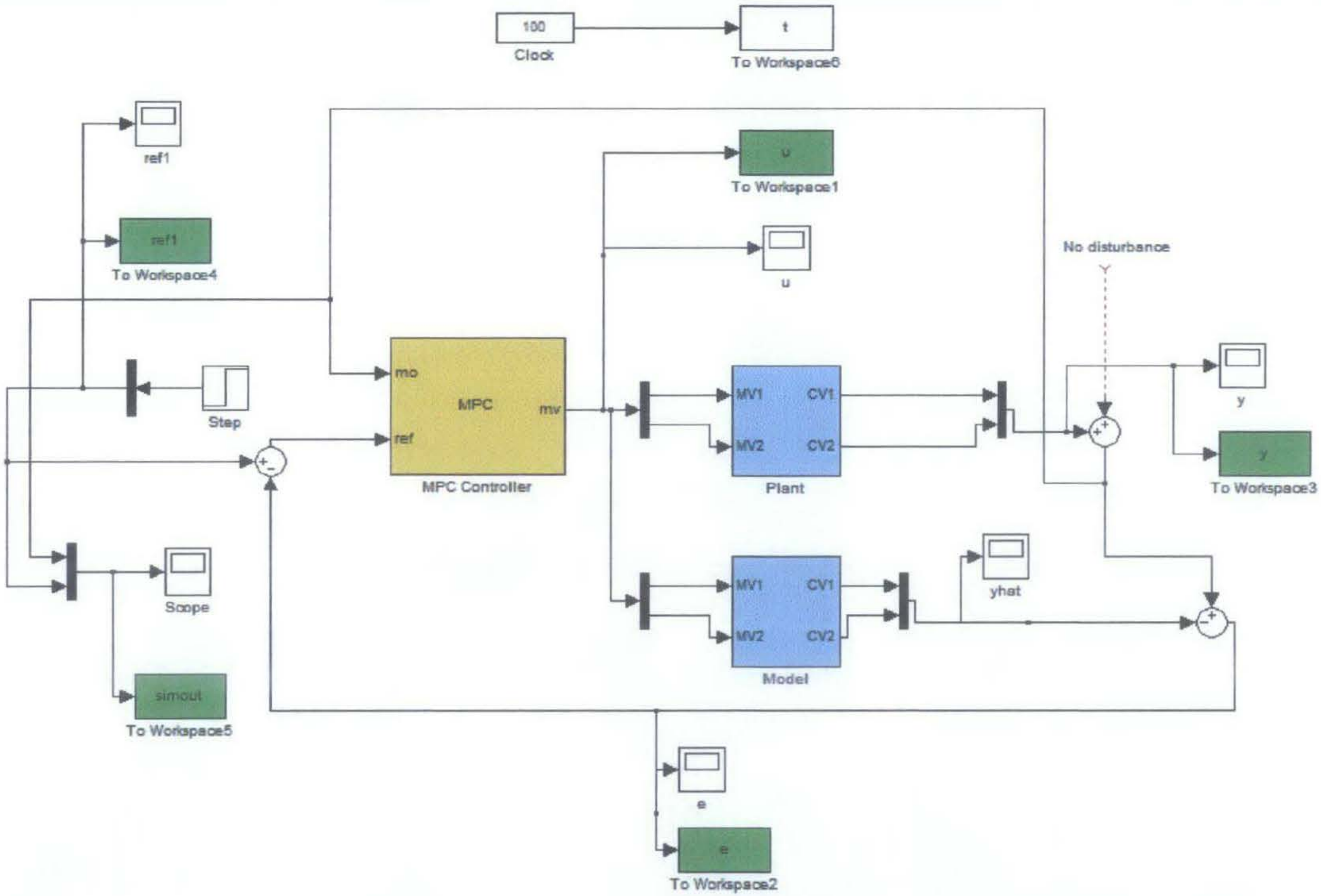


Figure 4.2: Wood and Berry distillation column simulation diagram in Simulink.

4.4 Taguchi Method of experimenting

Taguchi methods are statistical methods developed by Genichi Taguchi to improve the quality of manufactured goods, and more recently also applied to engineering.

Looking back into Figure 4.1, it is clear that each controlled variable is affected 'directly' by 2 of the 4 process models. CV1 is affected by G11 and G21 (with gain k11 and k21 respectively), while CV2 is affected by G12 and G22 (with gain k12 and k22 respectively). To understand more about the effects of each gain of the models on the outputs, we design a Taguchi experiment using the L25 Series, which is shown in the table below. Since we have 4 gains only, we use 4 factors from the table and the numbers 1 to 5 represents 5 different levels or changes of the gain.

Table 4.2: Table showing the Taguchi's L25 Series for 4 factors only.

	Factors			
Experiment	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	1	4	4	4
5	1	5	5	5
6	2	1	2	3
7	2	2	3	4
8	2	3	4	5
9	2	4	5	1
10	2	5	1	2
11	3	1	3	5
12	3	2	4	1
13	3	3	5	2
14	3	4	1	3
15	3	5	2	4
16	4	1	4	2
17	4	2	5	3
18	4	3	1	4
19	4	4	2	5
20	4	5	3	1
21	5	1	5	4
22	5	2	1	5
23	5	3	2	1
24	5	4	3	2
25	5	5	4	3

For the time being, positive and negative increments of 10% and 20% on the original gain will be used for the 5 different levels of the Taguchi experiment. Table 4.3 and Table 4.4 summarize the Taguchi experiment for the Wood and Berry distillation column case.

Table 4.3: Table showing the different gain values for the experiment.

% mismatch	-20%	-10%	0%	10%	20%
Levels	**1**	**2**	**3**	**4**	**5**
k11	10.24	11.52	12.8	14.08	15.36
k21	5.28	5.94	6.6	7.26	7.92
k12	-15.12	-17.01	-18.9	-20.79	-22.68
k22	-15.52	-17.46	-19.4	-21.34	-23.28

Table 4.4: Table showing the Taguchi's L25 Series according to 4 gain values.

	Factors			
Experiment	k11	k12	k21	k22
1	10.24	-15.12	5.28	-15.52
2	10.24	-17.01	5.94	-17.46
3	10.24	-18.9	6.6	-19.4
4	10.24	-20.79	7.26	-21.34
5	10.24	-22.68	7.92	-23.28
6	11.52	-15.12	5.94	-19.4
7	11.52	-17.01	6.6	-21.34
8	11.52	-18.9	7.26	-23.28
9	11.52	-20.79	7.92	-15.52
10	11.52	-22.68	5.28	-17.46
11	12.8	-15.12	6.6	-23.28
12	12.8	-17.01	7.26	-15.52
13	12.8	-18.9	7.92	-17.46
14	12.8	-20.79	5.28	-19.4
15	12.8	-22.68	5.94	-21.34
16	14.08	-15.12	7.26	-17.46
17	14.08	-17.01	7.92	-19.4
18	14.08	-18.9	5.28	-21.34
19	14.08	-20.79	5.94	-23.28
20	14.08	-22.68	6.6	-15.52
21	15.36	-15.12	7.92	-21.34
22	15.36	-17.01	5.28	-23.28
23	15.36	-18.9	5.94	-15.52
24	15.36	-20.79	6.6	-17.46
25	15.36	-22.68	7.26	-19.4

The result graphs of the 25 experiment simulations are shown in the APPENDIX. Figure 4.3 shows the result for experiment 25 only. After obtaining the above results for the 25 Taguchi experiments, Analysis of Means (ANOM) and Analysis of Variances (ANOVA) are carried out to determine the magnitude of the major contributors from the 4 gains to the overall process. Figure 4.4 and Figure 4.5 shows the results of ANOM contribution effects and ANOVA 'C_k' values respectively.

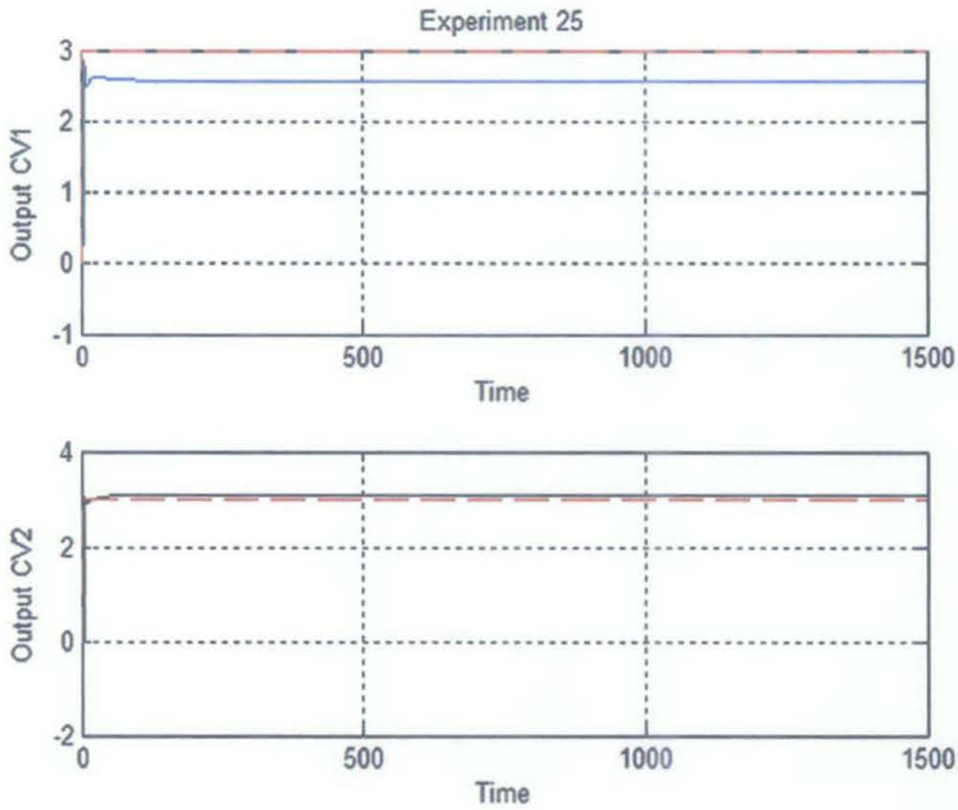


Figure 4.4: Taguchi result for experiment 25.

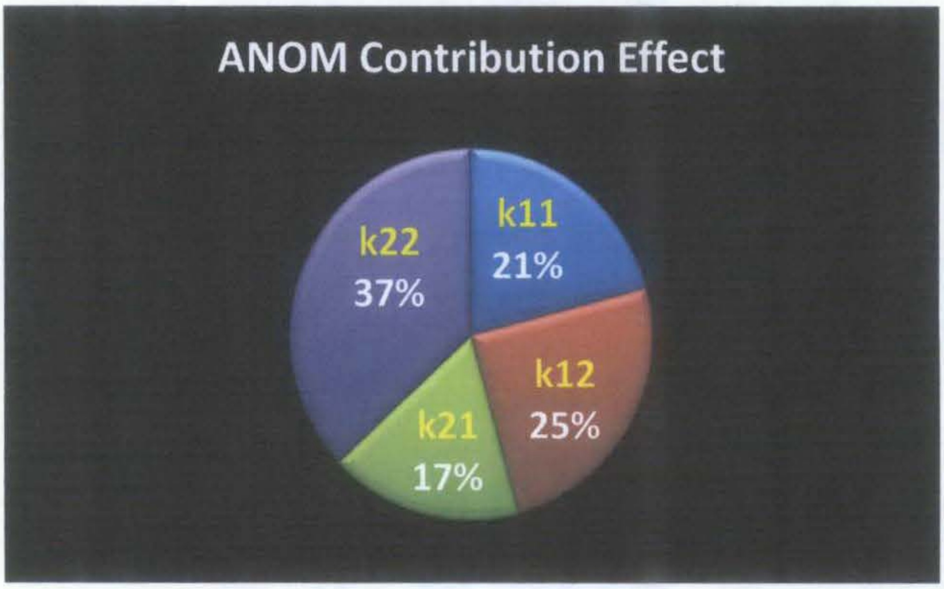


Figure 4.5: ANOM contribution effect distribution from Taguchi results.

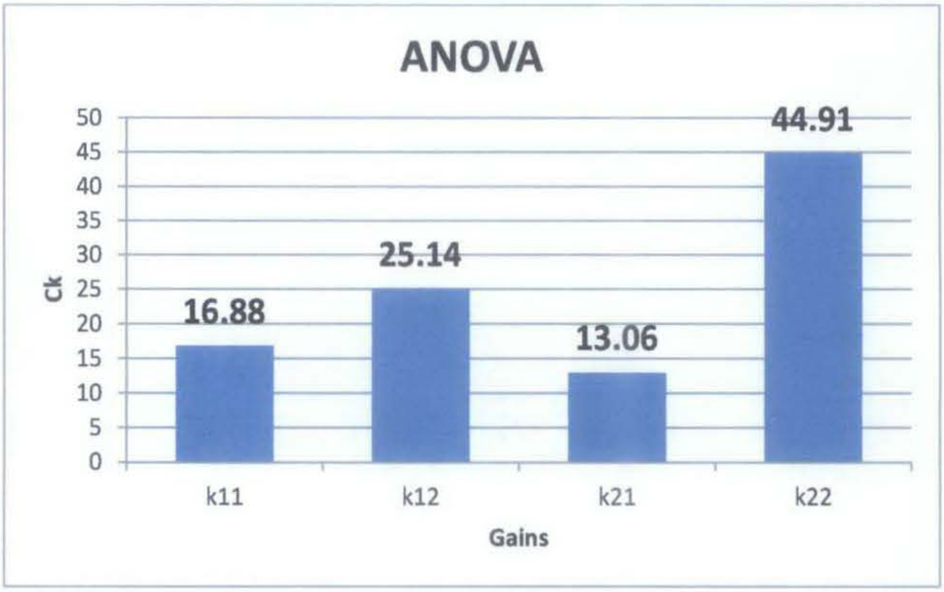


Figure 4.6: ANOVA C_k values distribution from Taguchi results.

Both the ANOM and ANOVA results agree with one another that k_{22} is the gain that contributes most to the process, and k_{12} is the second major contributor, followed by k_{11} and lastly k_{21}.

The summary of the conclusion and finding from this Taguchi experiment are the following:

- Since CV1 is affected directly by k11 and k12, ANOM and ANOVA suggests that k12 has greater effect.
- CV1 is largely affected by changes in k12 where a level of $\pm 20\%$ mismatch for this gain affects the response to offset far from the desired set point.
- Since CV2 is affected directly by k21 and k22, ANOM and ANOVA suggests that k22 has greater effect.
- CV2 is largely affected by changes in k22 where a level $\pm 20\%$ for this gain affects the response to offset far from the desired set point.
- Gains k12 and k11 have negligible effect on CV2, while gains k21 and k22 have negligible effects on CV1.

4.5 Finding the Threshold Limit

After having done the Taguchi experiments and getting the conclusions from it, the next task is to determine the specific and exact points of 'increase' or 'decrease' of each particular gain that will result in an intolerable output result. These points are called the threshold points. We will proceed one by one to find the threshold points both above and below the optimum gain.

The methodology for obtaining the threshold point is as follows:

- The plant gain model is increased (or decreased) incrementally and the set-point tracking response of the output is observed.
- According to Seborg (2004), a system is said to have reached its stable state when it oscillates 5% above or below the set point. Thus we take 5% as the maximum allowable offset for the output response.
- If the response output oscillates at greater than 5% at the end of the simulation, that incremental increase (or decrease) is recorded as the threshold point for the particular gain under observation.

As concluded from the Taguchi experiments that k12 affects CV1 significantly, therefore the response of CV1 is to be observed for changes of k12 in the positive and negative directions. Similarly, k22 affects CV2 significantly, therefore the response of CV2 is to be observed for changes of k22 in the positive and negative directions. Even though from the Taguchi experiments, it was clear that k11 and k21 have no significant roles on the output responses alone, they may have some contributions when combined with other gain mismatches and therefore their effects are monitored against CV1 and CV2 respectively.

k11 threshold limits

Figure 4.7 shows the results of Simulink simulations for positive increases of k11 from 0 to 90% mismatch for a set-point change of 3. The table for the different k11 values used is shown in the APPENDIX.

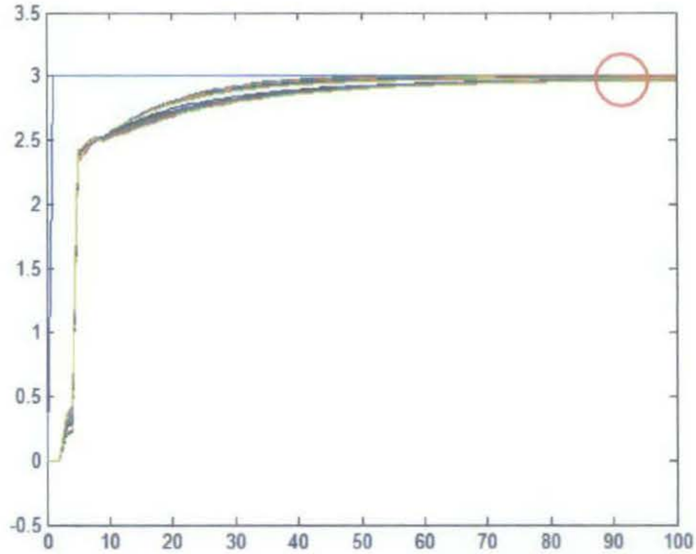


Figure 4.7: Output responses of CV1 for changes of k11 from 0 to 90%.

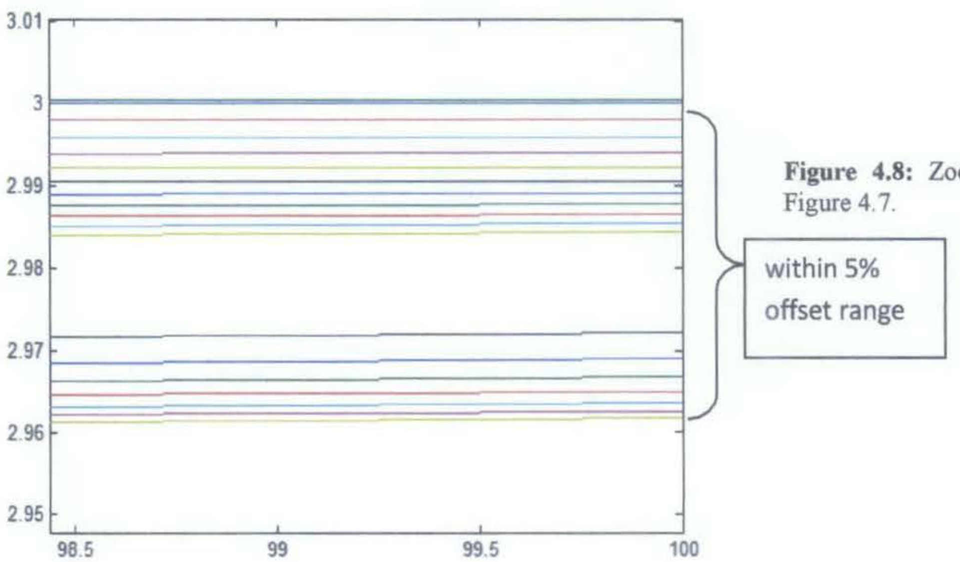


Figure 4.8: Zoomed view of Figure 4.7.

The results show that, even an increase of 90% mismatch of k11 has no significant effect on the output CV1 where it settles into the new set-point of 3 (with less than 5% difference).

Figure 4.9 shows the results of Simulink simulations for **negative** decreases of k11 from 0 to -18% mismatch for a set-point change of 3. The table for the different k11 values used is shown in the APPENDIX.

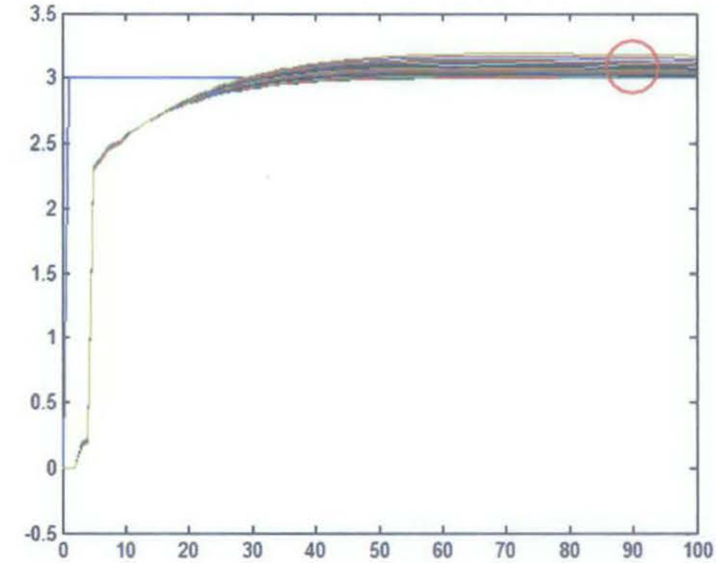


Figure 4.9: Output responses of CV1 for changes of k11 from 0 to -18%.

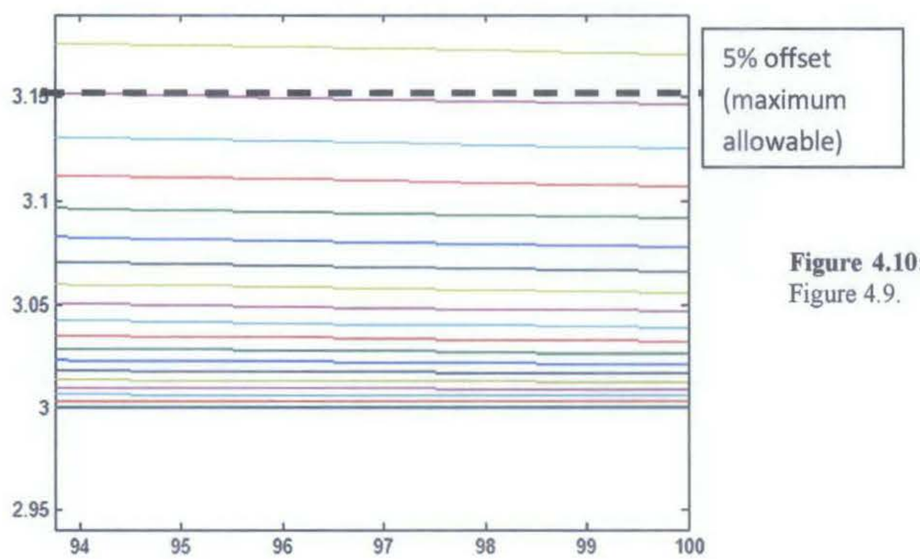


Figure 4.10: Zoomed view of Figure 4.9.

The results show that, a decrease of -18% mismatch of k11 has significant effect on the output response, causing the offset to be greater than the maximum allowable 5%, and thus is considered the threshold limit for k11 mismatch in the negative direction.

Threshold limit for k11:	**-18%** to **90%**

Following the same procedure for k12, k21 and k22, we obtain the following threshold limits for each gain:

Table 4.5: Summary of threshold limits for mismatches.

Gain	Min allowable mismatch(%)	Max allowable mismatch(%)
k_{11}	-18	90
k_{12}	-6	5
k_{21}	-90	35
k_{22}	-6	4

The values of the different k12, k21 and k22 values used for determining the threshold values are shown in the APPENDIX.

A comparison between our threshold limits for the gains and ANOM and ANOVA from Taguchi, shows that the greater the contribution of the gain, the more critical and narrow is its threshold limit.

4.6 **Partial Correlation Analysis** - Qualitative approach

To relate the threshold limit method of detecting model plant mismatch (which is the objective of this project), a method for partial correlation analysis detection of mismatch is needed to be developed in the simulations. There exist multiple methods of calculating partial correlations, one possible method is (Carlsson, 2010),

$$B_y = ((\mathbf{z}^T\mathbf{z})^{-1}\mathbf{z}^T\mathbf{y})$$

$$\mathbf{r_y} = \mathbf{y} - \mathbf{z}B_y$$

$$B_x = ((\mathbf{z}^T\mathbf{z})^{-1}\mathbf{z}^T\mathbf{x})$$

$$\mathbf{r_x} = \mathbf{x} - \mathbf{z}B_x$$

$$ParCorr\ (\mathbf{x},\mathbf{y}|\mathbf{z}) = Corr(\mathbf{r_x},\mathbf{r_y})$$

The MATLAB coding for obtaining the partial correlation plots is shown in the APPENDIX. The results obtained for k11 is shown in Figure 4.11 which detects the 50% mismatch correctly.

As can be seen from Figure 4.11, that the partial correlation analysis method proposed by Badwe et. al (2009) shows when does a mismatch occur qualitatively by seeing how many red sticks in the plot, but it does not convey a quantitative information about the extent of the mismatch or the effect on the performance of the MPC on the process.

Therefore, the next section attempts to derive a method that can relate the partial correlation method to detecting the threshold point of mismatch for each gain.

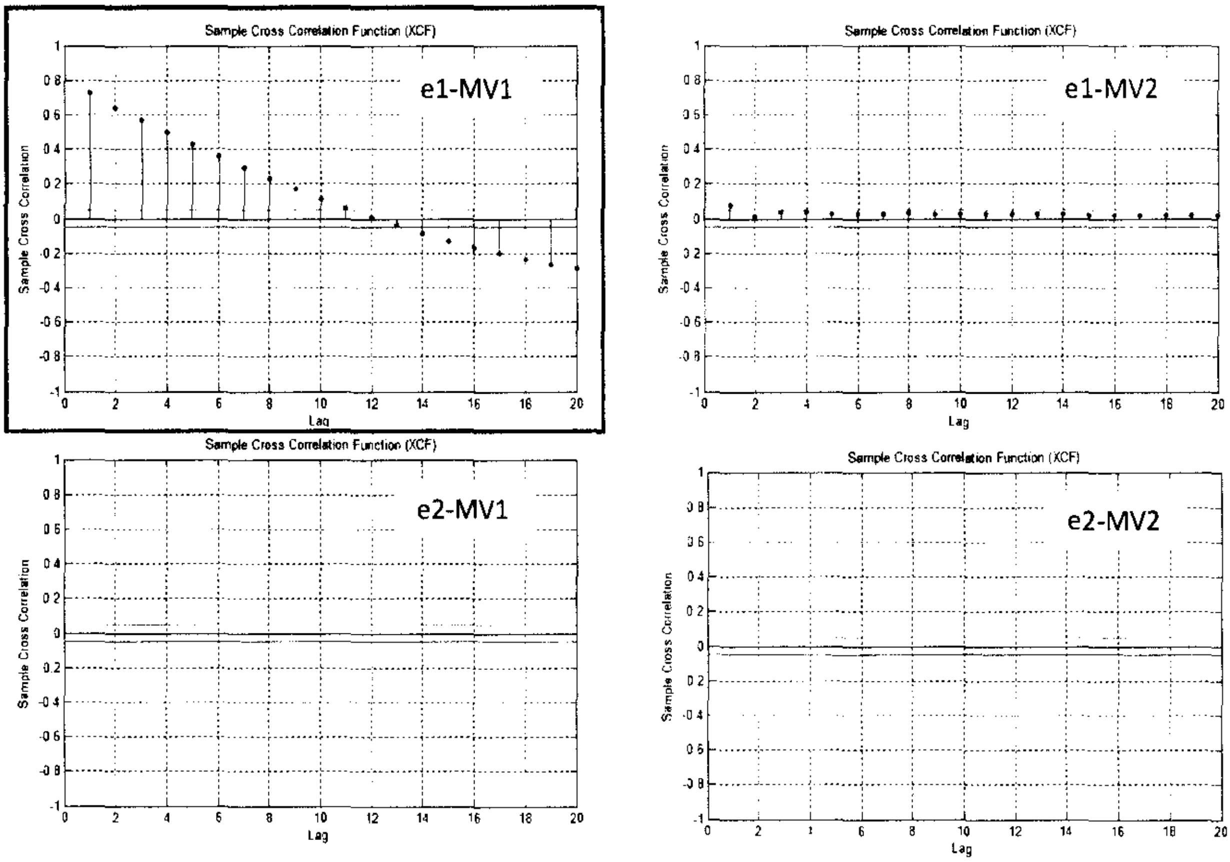


Figure 4.11: Partial correlation plots detecting 50% mismatch in k11 correctly.

4.7 Finding an indicator variable for plant gain

Due to the shortcomings of the partial correlation method in informing responsible operators or supervisors in making decisions on the quality of the model being used, the proposed method is desired to utilize the threshold limit points for the mismatch and alarm operators automatically when an intolerable mismatch situation occurs.

Since we have determined the threshold points, our aim is to find an indicator variable to tell us when we have crossed the threshold limit. The methods are:

I. *Gain vs Partial Correlation Coefficient* method
II. *Gain vs Mean Error* method
III. *Gain vs (e/u)* method

The next section explains each method with any advantage or disadvantage it possesses.

4.7.1 Gain vs Partial Correlation Coefficient method

A plot of the k11 gain values versus the partial correlation coefficient yields the following:

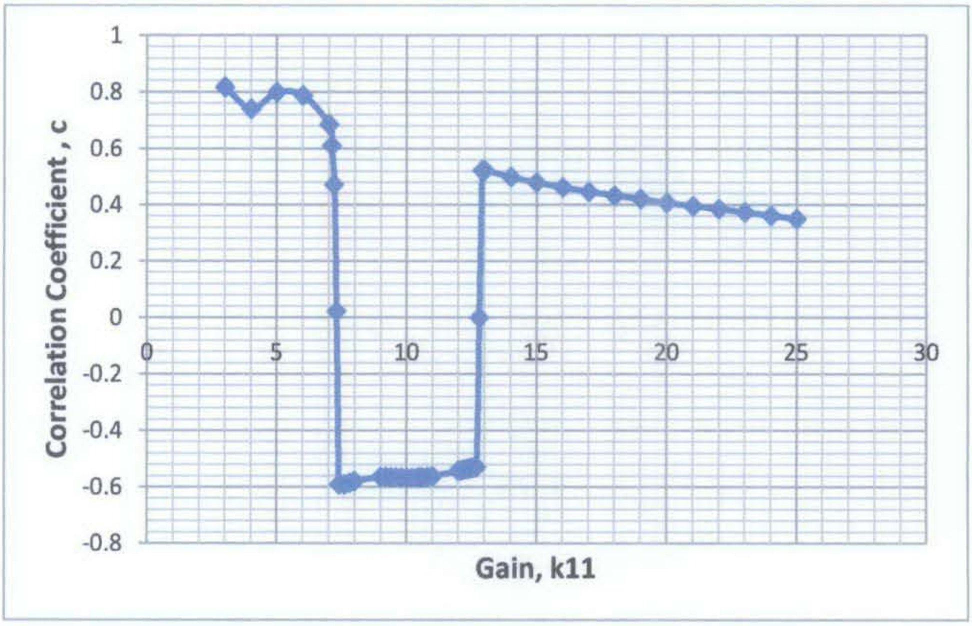


Figure 4.12: Graph of gain values versus the partial correlation coefficient.

The resulting graph in Figure 4.12 shows a very interesting relationship. However, there are few problems with this graph. First, it is difficult to fit such a curve into any sort of regression fitting equations. Second, is that the graph is somehow symmetrical for negative values of the correlation coefficient, meaning for a single value of partial correlation coefficient obtained from analysis, there can be 2 values of gains, where one might be a good estimate while the other being heavily mismatched and needs re-identification. This problem makes it difficult for operators and supervisors to make a decision because the exact value of the gain is actually unknown.

4.7.2 Gain vs Mean Error method

A plot of the k11 gain values versus the mean error yields the following:

Figure 4.13: Graph of gain values versus the mean error.

The result is an exponential curve that is easy to fit and non-symmetrical, meaning each value of the mean error denotes one specific Gain value. So if we can calculate the mean error and compare it with the mean error of the threshold point mean error, we can know whether the gain have exceeded the allowable mismatch limit or not. However, the disadvantage of this method is that mean error is not a fixed variable which changes with changing set-point and any other variable. Therefore, we do not have a standard method for detecting the threshold limit yet.

4.7.3 Gain vs (e/u) method

A plot of the k11 gain values versus the average mismatch which is the average residual error divided by the average manipulated variable yields the following:

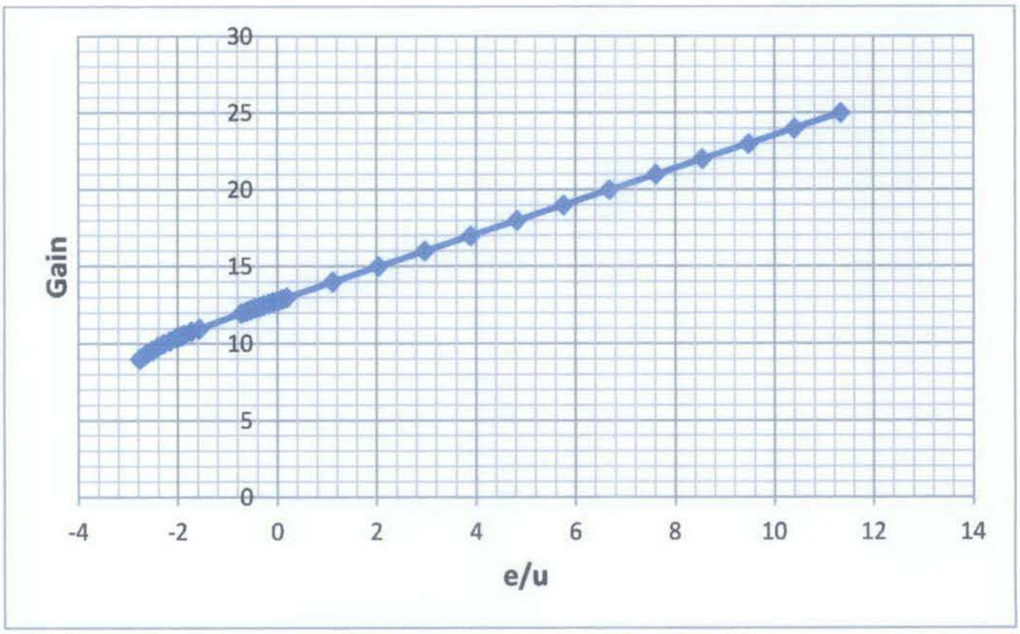


Figure 4.14: Graph of gain values versus e/u values.

Interestingly it yields almost a linear graph for k11 gain. The advantage of this method is that the variable (average residual/average MV) is constant for different set-point change scenarios over some range. This makes it more universal and also each value corresponds to one gain value. The red line in the figure shows the threshold gain obtained earlier at 10.5 for k11 which corresponds to -1.9621 of (average residual/average MV). So, from the above graph we see that any value of (average residual/average MV) less than -1.9621 can be considered outside the threshold limit and serious action must be taken.

The above e/u relationship is derived from Badwe et al. (2010):

$$e = u\Delta \longrightarrow \frac{e}{u} = \Delta$$

where

e	:	*model residual*
u	:	*manipulated variables*
Δ	:	*model mismatch*

4.8 Auto-detection and auto-correction - Quantitative approach

Using method III of the previous section to detect mismatch in plant model, any changes of the gain in the plant model can be identified at any point in time. First, a plot of (gain vs e/u) for all 4 gains in the case study is plotted as shown in Figure 4.15.

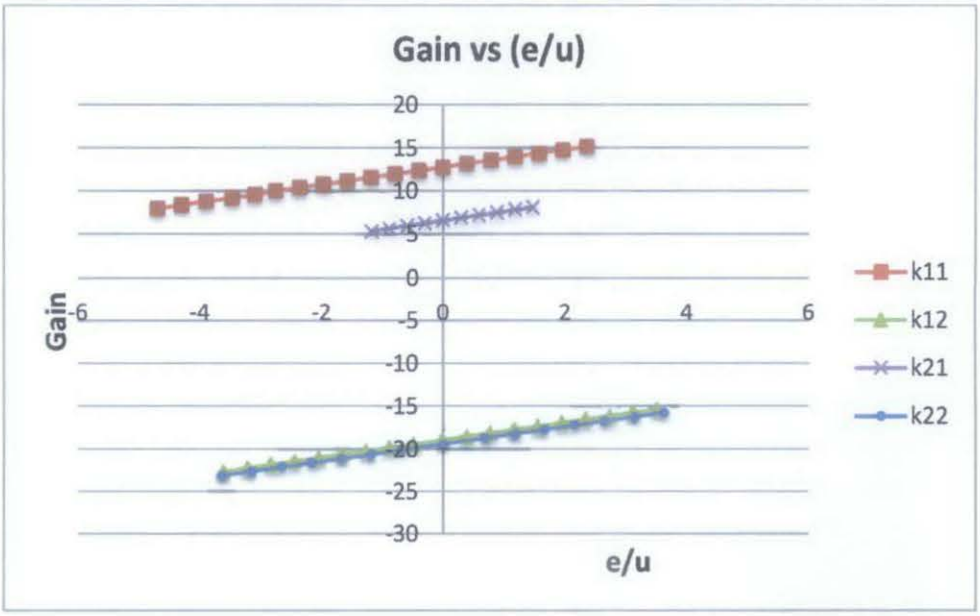


Figure 4.15: Graph of gain values versus the average mismatch.

The linear fit of the straight lines obtained in Figure 4.11 for the different gain vs (e/u) relationship are:

$k_{11} = 1.0231(e_1/u_1) + 12.795$ $R^2 = 0.9999$
$k_{12} = 1.025(e_1/u_2) - 18.9$ $R^2 = 1$
$k_{21} = 1.0162(e_2/u_1) + 6.6001$ $R^2 = 1$
$k_{22} = 1.0183(e_2/u_2) - 19.399$ $R^2 = 1$

The R^2 values clearly indicate that a very linear relationship is obtained. With the above equations we only need to measure '**e**' and '**u**' values from time to time and calculate the gain values for the plant model. If the gain value has changed and crossed the threshold limit, the corresponding gain in the controller Model is changed to the same value of the plant gain, tracking it.

This method uses measurable values from the real plant operation and compares it with threshold limits values. Therefore, this is a quantitative method that represents the extent of mismatch in a more practical way.

To summarize the proposed auto-correction method procedure:

1. The value (e/u) is first calculated.
2. Using the above equations, the gain value is estimated.
3. The estimated gain is compared with the corresponding model gain and corrected if out of the limit.

CHAPTER 5:
VALIDATING THE RESULTS

To validate the results obtained, we need to check the proposed methodology and algorithm on our case studies. If the proposed methodology can detect an intolerable mismatch and correct it automatically, then our results can be said to be validated.

5.1 Plant simulation conditions

For our validating purpose, our plant simulation using MATLAB Simulink will follow the following conditions:

Simulation Time: 400 min

Step input: 1 to 10

Disturbance: 0 (no disturbance)

k_{11} *mismatch:* 0 %

k_{12} *mismatch:* +20 %

k_{21} *mismatch:* 0 %

k_{22} *mismatch:* -20 %

The reason for using a simulation time of 400 min is because the proposed method has been tested to be effective in detecting the gain mismatch when sufficient data of the model residuals $\boldsymbol{e}$ and manipulated variables $\boldsymbol{u}$ has been collected. For the current project, a detection time of 200 min was selected that gives satisfying accuracy of results. The longer the time the better is the accuracy but the disadvantage in the fact that a longer time will be needed before a mismatch will be detected and corrected if present. Further study and analysis may reduce the set time of 200 min using this proposed approach, but for the purpose of initial validation the testing was proceeded with 200 min.

Step inputs of 1 to 10 were tested for validation of the robustness of the proposed methodology with various conditions of input. However, at the moment the study is being carried out for set-point tracking only and the assumption of no disturbance holds.

For the gain values, mismatch is intentionally introduced into only 2 of the 4 gains in the system, namely k_{22} and k_{12} which were ranked earlier as most and second most contributing gains to the overall process. The reason for this choice is because of the proposed method can detect changes in one only plant gain out of the 2 gains affecting directly each output. Detecting simultaneous mismatches in both gains affecting the same output is currently not possible. Therefore, a compromise for one of the gains is needed, and based on the taguchi results and threshold limits earlier, the more tight gain is selected for each pair. However, simultaneous mismatches in the gains affecting the different outputs can be detected by the proposed methodology and thus the 2 gains that are mismatched in the validation process are k_{22} and k_{12} assuming that if these two are corrected, the rest have less impact.

5.2 Validation results

The result of the validation testing is shown in the figures shown. Figure 5.1 shows the result of -10% mismatch for both k_{12} and k_{22} gains. Figure 5.2 shows the result of -20% mismatch for both k_{12} and k_{22} gains. Figure 5.3 shows the result of +10% mismatch for both k_{12} and k_{22} gains. Figure 5.4 shows the result of +20% mismatch for both k_{12} and k_{22} gains.

For each of the figures, two plots are displayed one for each of the two process outputs; CV1 and CV2. The desired set point is also shown in red line for both figures. The auto-correction algorithm is run at time = 200 min.

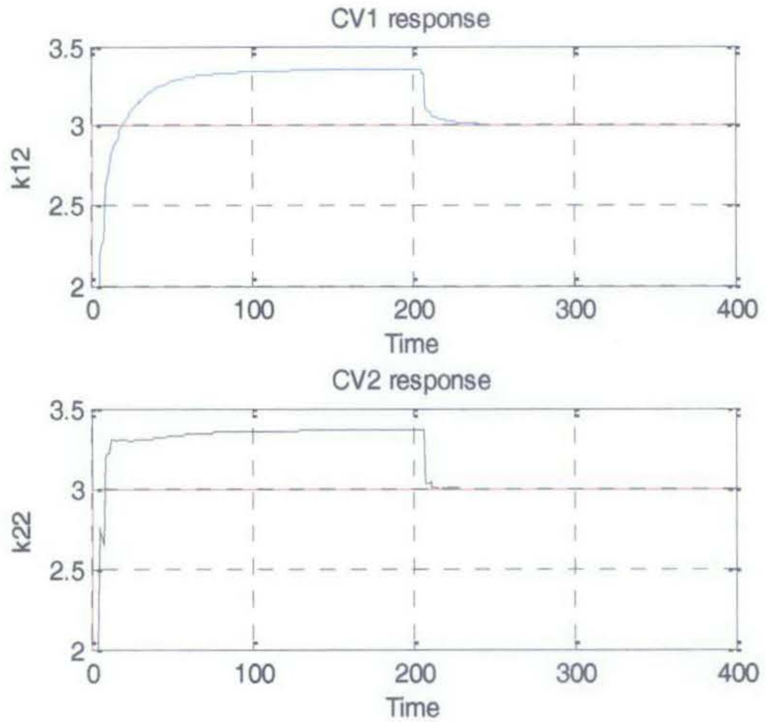


Figure 5.1: CV1 and CV2 responses for -10% mismatches of k_{12} and k_{22}.

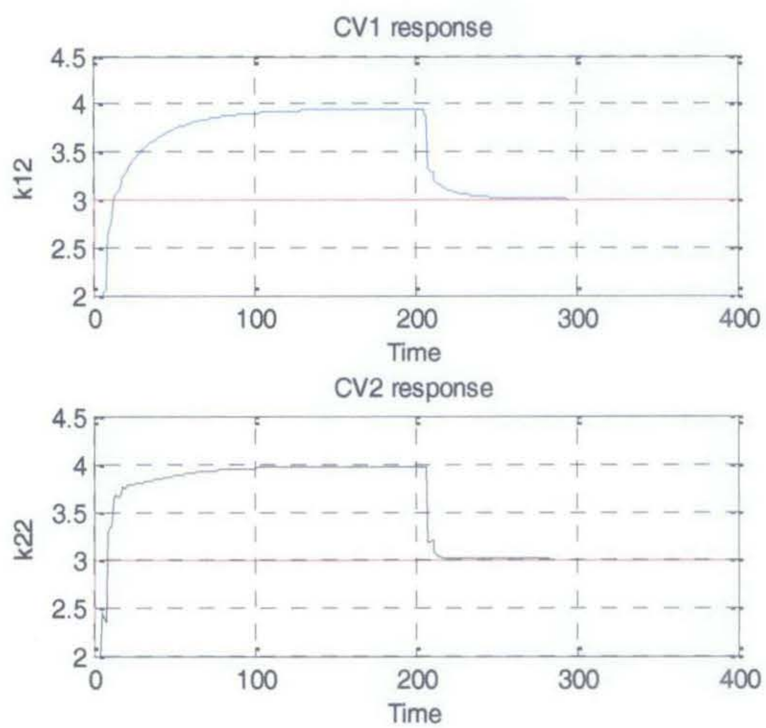


Figure 5.2: CV1 and CV2 responses for -20% mismatches of k_{12} and k_{22}.

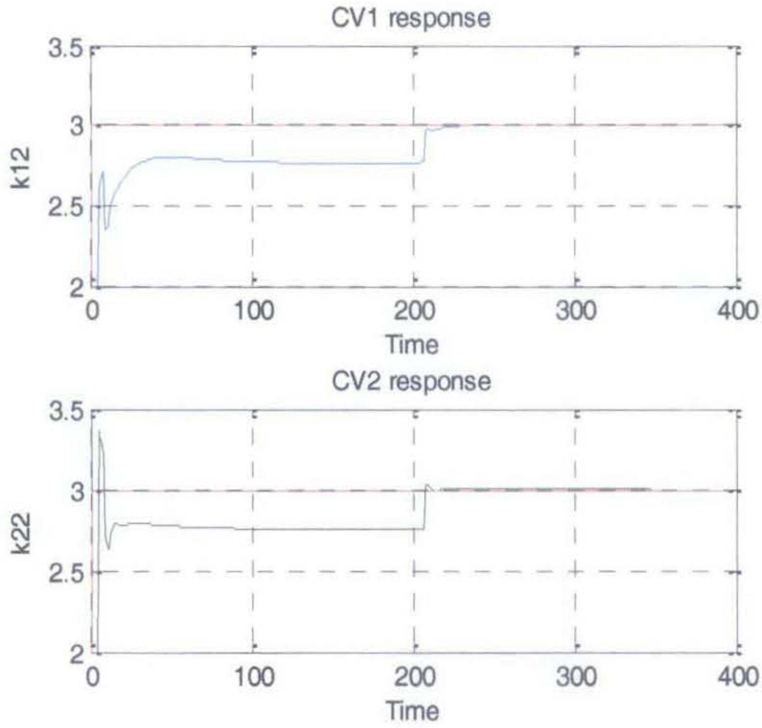


Figure 5.3: CV1 and CV2 responses for +10% mismatches of k_{12} and k_{22}.

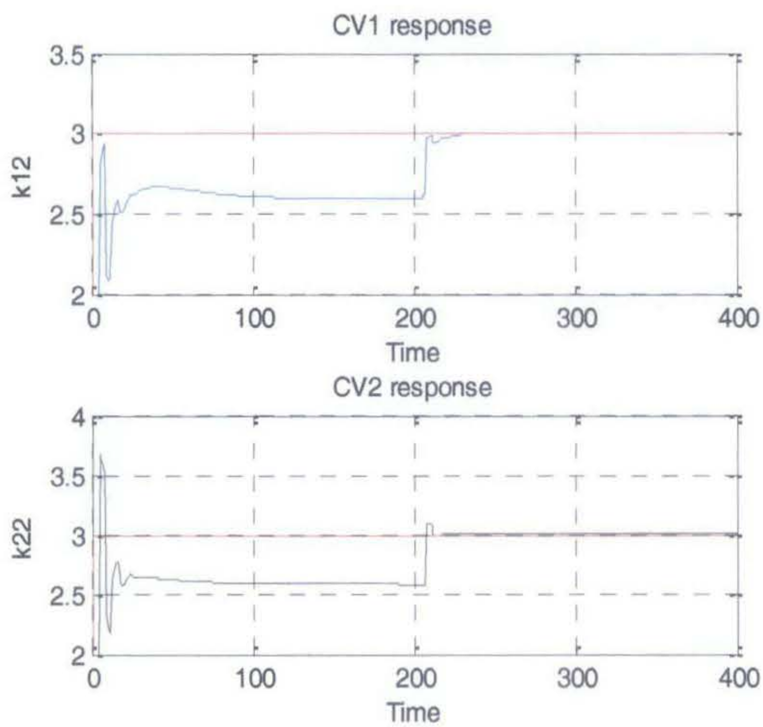


Figure 5.4: CV1 and CV2 responses for +20% mismatches of k_{12} and k_{22}.

5.3 Efficiency of the proposed method

From the results displayed in the previous figures, it is clear that the proposed methodology for auto-detection and auto-correction have been validated. For all 4 validation tests for the different extents of mismatches initiated simultaneously in both k_{22} and k_{12}, the controller was able to detect the mismatch almost accurately and successfully corrected it to the set point level.

Summary of the auto-detection and auto-correction method:

- It requires enough time of 200 min to collect sufficient data for detection and then correction.
- It takes a very short time for correction action to bring output back to set-point, in the range of 10-30 min.
- Accurate prediction of gain values even under simultaneous changes or mismatches of the two major contributor gains to the process; which are k_{22} and k_{12}.

The algorithm coding for carrying out the proposed methodology of auto-detection and auto-correction is shown in Figure 5.5.

```
%Sorting error and manipulated variables for each line
e1=e(:,1);
e2=e(:,2);
u1=u(:,1);
u2=u(:,2);

%Calculating e/u values
k_11=sum(e1)/sum(u1);
k_12=sum(e1)/sum(u2);
k_21=sum(e2)/sum(u1);
k_22=sum(e2)/sum(u2);

%Predicting gain values using linear equations obtained earlier
k11=(1.0998*(k_11))+12.765;
k12=(1.1346*(k_12))-18.908;
k21=(1.0552*(k_21))+6.6012;
k22=(1.0989*(k_22))-19.389;

%Detecting k12 mismatch

        if (k12)<(-17.96)

       if (k12)>(-19.85)
        disp('k12=No Mismatch')

        else
        disp('k12=MISMATCH')

        set_param('set_point_tracking/Model/G12','Numerator','k12')
        disp('k12=CHANGED')

       end

     else
     disp('k12=MISMATCH')

     set_param('set_point_tracking/Model/G12','Numerator','k12')
    disp('k12=CHANGED')

    end

%Detecting k22 mismatch

    if (k22)<(-18.43)

       if (k22)>(-20.37)
        disp('k22=No Mismatch')

        else
        disp('k22=MISMATCH')

        set_param('set_point_tracking/Model/G22','Numerator','k22')
        disp('CHANGED')

       end
```

```
    else
     disp('k22=MISMATCH')

     set_param('set_point_tracking/Model/G22','Numerator','k22')
    disp('k22=CHANGED')

    end


%Detecting k11 mismatch

    if (k11)<(24)

       if (k11)>(10.5)
        disp('k11=No Mismatch')

        else
        disp('k11=MISMATCH')

       end

    else
     disp('k11=MISMATCH')

    end


%Detecting k21 mismatch

    if (k21)<(8.9)

       if (k21)>(1)
        disp('k21=No Mismatch')

        else
        disp('k21=MISMATCH')

       end

    else
     disp('k21=MISMATCH')

    end

%END
```

Figure 5.5: Algorithm coding for detection and correction of mismatches of k_{12} and k_{22}.

CHAPTER 6:
CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

Detection of model plant mismatch exact location is very important in order to save time and money especially since plant is made up of big systems with so many models. The current available detection methods have two shortcomings. First, it works based on a pre-specified deviation limit of 5% which may not be necessary. Processes may be able to tolerate more and thus a re-identification of the system is not justified at that level. Second, to identify the mismatch, manual tabulation and plotting need to be done and observed to come up with the interpretation.

The proposed methodology solve the first shortcoming by using the help of Taguchi experiments to determine the exact threshold limit for each gain in the system and determine which is the most critical gains to the overall process, instead of assuming intolerable mismatch whenever any of the gains shifted from the original model value.

The proposed methodology solves the second shortcoming by detecting mismatch in the gain of a model quantitatively using the variable 'e/u' or model errors divided by manipulated variables, which was found to have a very linear relationship with the plant gain. This means that by measuring only 'e' and 'u' from the plant, the plant model gains can be estimated to great deal of accuracy. Using this estimate, the model gain of the controller is adjusted to the new plant gain values in what has been termed as an auto-correction procedure.

The proposed method was assessed in simulations using MATLAB and Simulink on the Wood and Berry distillation column case study and was successfully validated. Testing for various mismatch scenarios for both two major contributors to the process, the algorithm was able to bring the output back to the desired set-point in a very short time.

6.2 Recommendations for future work

The proposed methodology was all studied under a no disturbance assumption where the primary objective was set-point tracking. Next would be to assess and probably modify the proposed methodology under disturbance rejection conditions. Until the success of this stage, then only the project would proceed to the second case study; Shell heavy oil fractionators which will reinforce the validation of the proposed methodology.

Some of the limitations of the auto-detection method proposed that can be improved further is the need for a minimum of 200 min of time before any detection analysis or correction procedure taken. For some processes 200 min might be a very long time where many problems could happen to the plant by the time the controller takes action.

Another recommendation for further study into this topic is to solve the problem of simultaneous gain mismatches for all gains in the system, or more specifically of those gains affecting the same output directly. This will in turn make the reaction of the controller to mismatch more robust.

REFERENCES

Anderson, J. S. (1992). Process control opportunities and benefits analysis. *Proc. Advanced Control for the Process Industries.*

Badwe, A. S., Patwardhan, R. S., Shah, S. L., Patwardhan, S. C., & Gudi, R. D. (2010). Quantifying the impact of model-plant mismatch on controller performance. *Journal of Process Control, 20*(4), 408-425.

Badwe, A. S., Gudi, R. D., Patwardhan, R. S., Shah, S. L., & Patwardhan, S. C. (2009). Detection of model-plant mismatch in MPC applications. *Journal of Process Control, 19*(8), 1305-1313.

Badwe, A. S., Shah, S. L., & Patwardhan, S. C. (2008). Model-Plant Mismatch Detection in MPC Applications using Partial Correlation Analysis. *Proceedings of the 17th World Congress the International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.*

Kano, M., Shigi, Y., & Hasebe, S. (2010). Detection of significant model-plant mismatch from routine operation data of model-predictive control system. *Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems (DYCOPS 2010)*, 677-682.

Morari, M., & H. Lee, J. (1999). Model predictive control: Past, present and future. *Computers & Chemical Engineering, 23*(4-5), 667-682.

Nafsun, A. I., & Yusoff, N. (2011). Effect of model-plant mismatch on MPC controller performance. *Journal of Applied Sciences.*

Orukpe, P. E. (2005). Basics of model predictive control. *Imperial College, London.*

Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice, 11*(7), 733-764.

Qin, S.J. (1998). Control performance monitoring-review and assessment. *Computers and*

Chemical Engineering, 23 (2), pp.173-186.

Seborg, D. E., Edgar, & T.F. and Mellichamp, D. A. (2004). *Process dynamics and control* (2nd ed.). New York: Wiley.

Selvanathan, S., & Tangirala, A. K. (2010). Diagnosis of poor control loop performance due to Model Plant mismatch. *Industrial & Engineering Chemistry Research, 49*(9), 4210-4229.

Thornhill, N. F., Huang, B., & Shah, S. L. (2003). Controller performance assessment in set point tracking and regulatory control. *International Journal of Adaptive Control and Signal Processing, 17*(7-9), 709-727.

Wang, Y., & Wang, X., (2010). Performance diagnosis of MPC with model-plant mismatch. *Chinese Control and Decision Conference.*

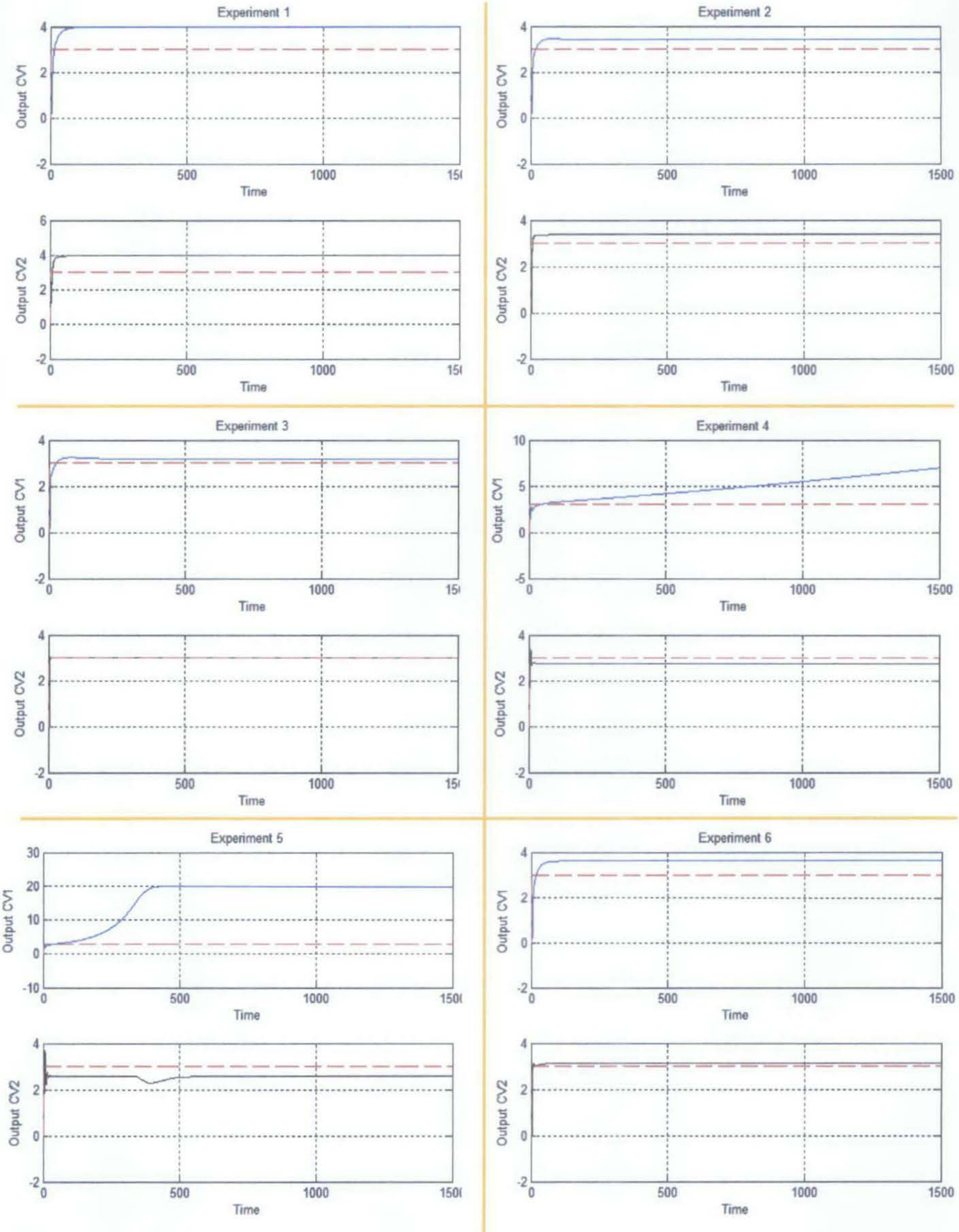
Willis, M., & Tham. M., (1994). Advanced process control and on-line optimisation in shell refineries. *Chemical Engineering and Advanced Materials.*

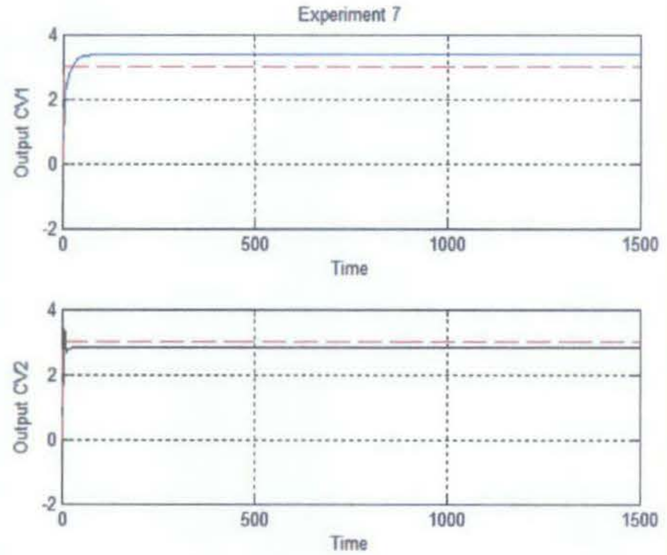
APPENDIX

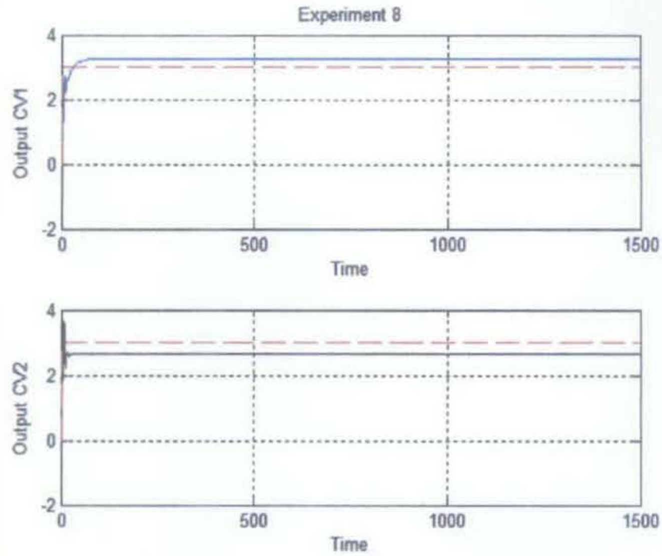
MPC design coding in MATLAB

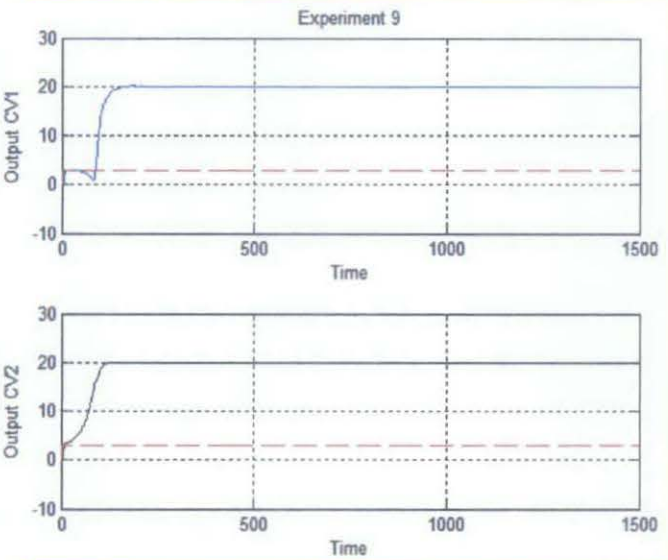
```
%%
% This demonstration shows MPC to control a multi-input multi-output
%system for Wood and Berry model. The system has 2 manipulated variables
and
% 2 measured outputs.
%%
Ts = 1; % sampling time (delt)
g11=tf(12.8, [16.7 1],'IOdelay',1);
g12=tf(-18.9, [21 1],'IOdelay',3);
g21=tf(6.6, [10.9 1],'IOdelay',7);
g22=tf(-19.4, [14.4 1],'IOdelay',3);
G1=[g11 g12;g21 g22];

G1d=c2d(G1,Ts);  % discrete-time model
G1ss=ss(G1d); % in SS form

%% Define process and disturbance model
model=(G1d);
Model.Plant= model;
%model.InputName={'MV1';'MV2';'MD1';'MD2'};
model.InputName={'MV1';'MV2'};
model.OutputName={'CV1';'CV2'};
% model.InputGroup.MV = 2; %2 inputs
% model.OutputGroup.MO = 2; %2 outputs
model = setmpcsignals(model, 'MV1', [1],'MV2',[2], 'MO1', [1],'MO2',[2]);

%model = setmpcsignals(model, 'MV1', [1],'MV2',[2], 'MO1',
[1],'MO2',[2],'MD1', [1],'MD2',[2]);

clear InputSpecs OutputSpecs % to clear the structure from previous work

% Constraints on inputs and outputs
InputSpecs(1)=struct('Min',-[],'Max',[],'RateMin',-10,'Ratemax',10);
InputSpecs(2)=struct('Min',-[],'Max',[],'RateMin',-10,'Ratemax',10);
OutputSpecs(1)=struct('Min',-20,'Max',20);
OutputSpecs(2)=struct('Min',-20,'Max',20);

% Weight on inputs and outputs
Weights=struct('ManipulatedVariables',[0 0],...
               'ManipulatedVariablesRate',[.1 .1],...
               'OutputVariables',[1 10]);

% Perition and control horizon
p=30; %prediction horizon
m=2; %control horizon

%MPC objective
MPC1=mpc(model,Ts,p,m,Weights,InputSpecs,OutputSpecs);

%MPC state
% xmpc=mpcstate(MPCobj,xp,xd,xn,u)
xmpc=mpcstate(MPC1);
```

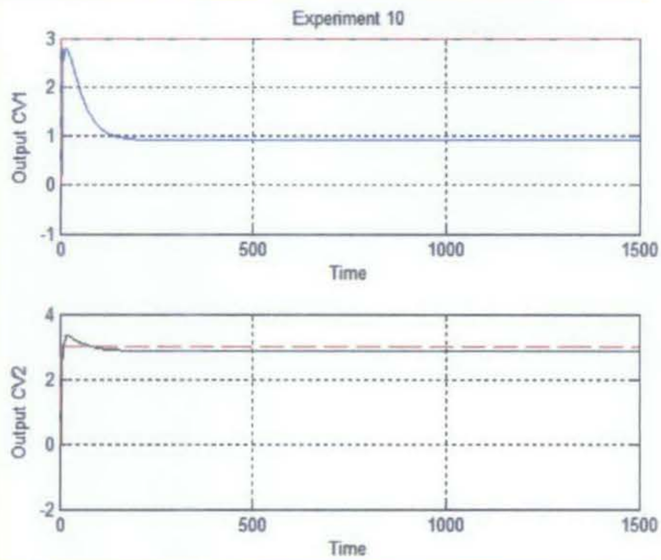
Taguchi Experiment results

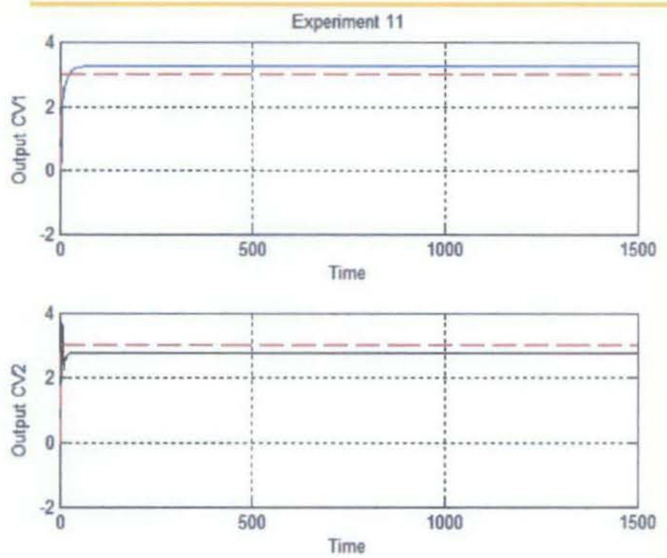


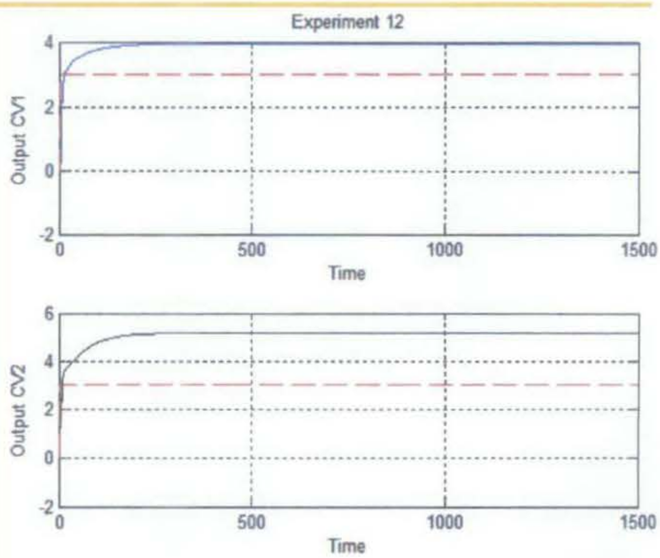

Experiment 7
Output CV1
Output CV2
Time

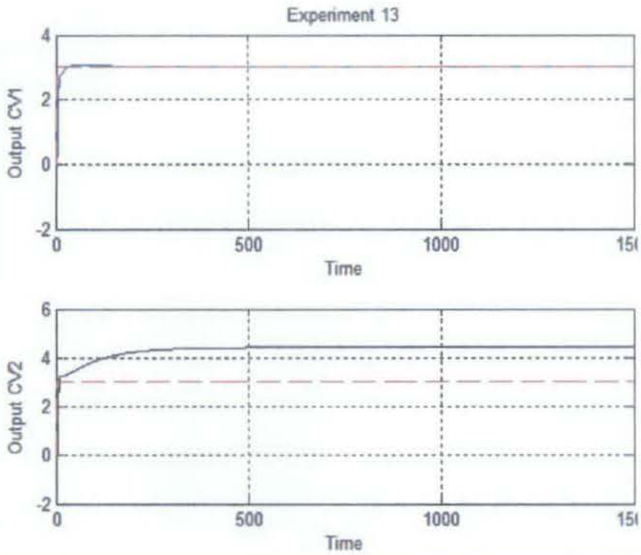

Experiment 8
Output CV1
Output CV2
Time

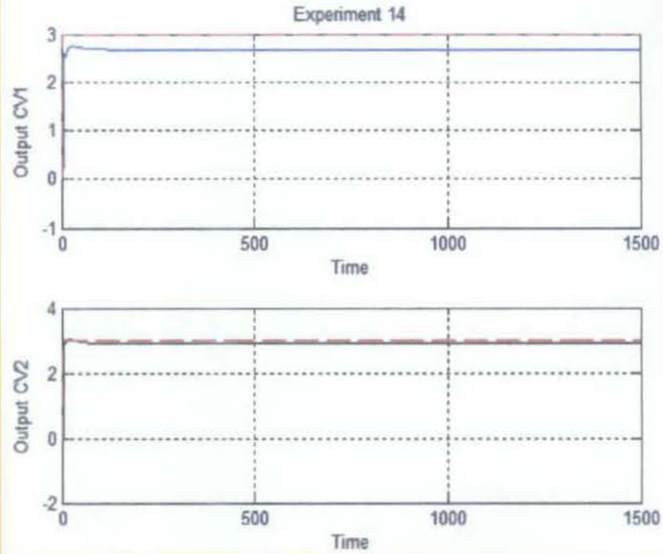

Experiment 9
Output CV1
Output CV2
Time

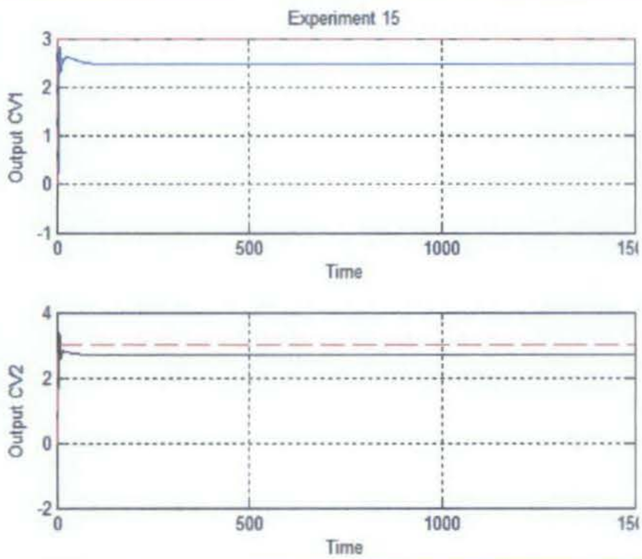

Experiment 10
Output CV1
Output CV2
Time

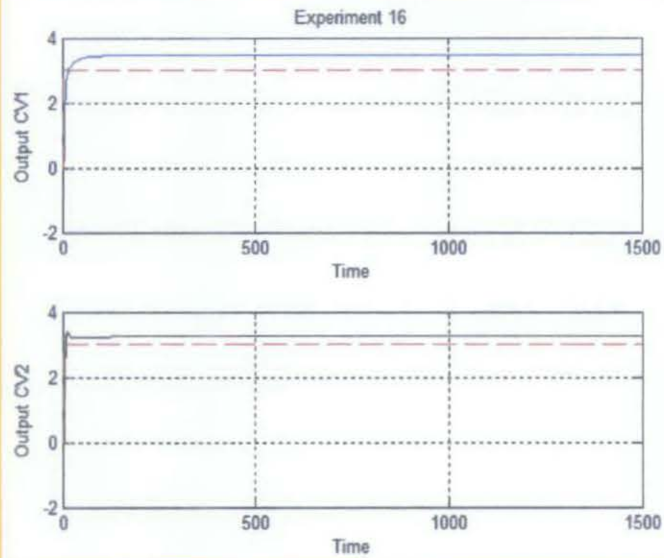

Experiment 11
Output CV1
Output CV2
Time

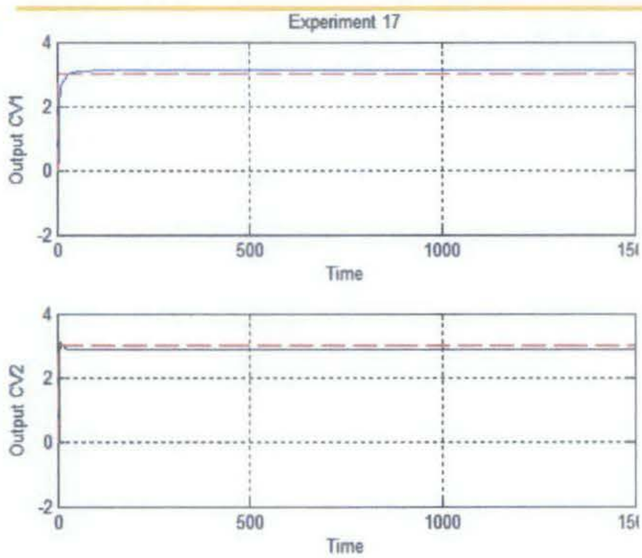

Experiment 12
Output CV1
Output CV2
Time

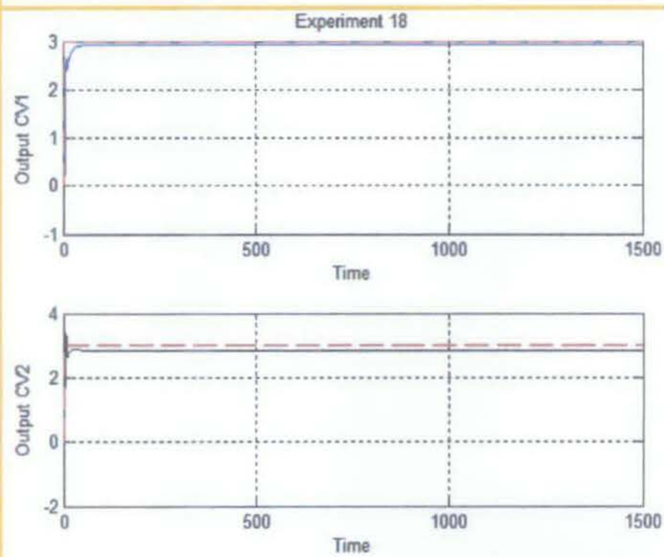

Experiment 13
Output CV1
Output CV2
Time
Time

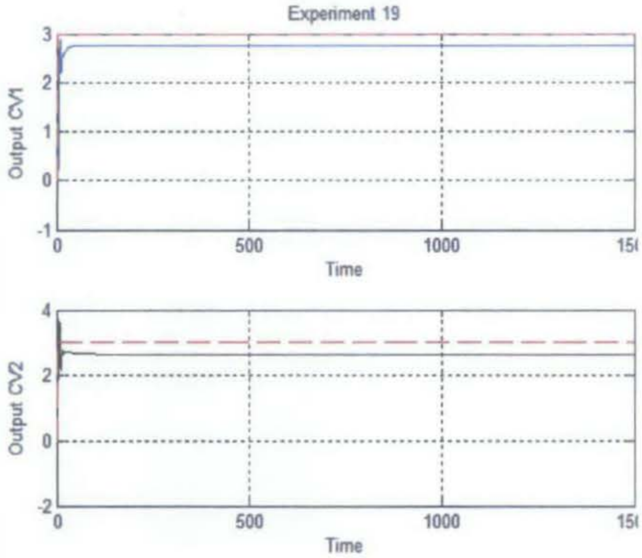

Experiment 14
Output CV1
Output CV2
Time
Time

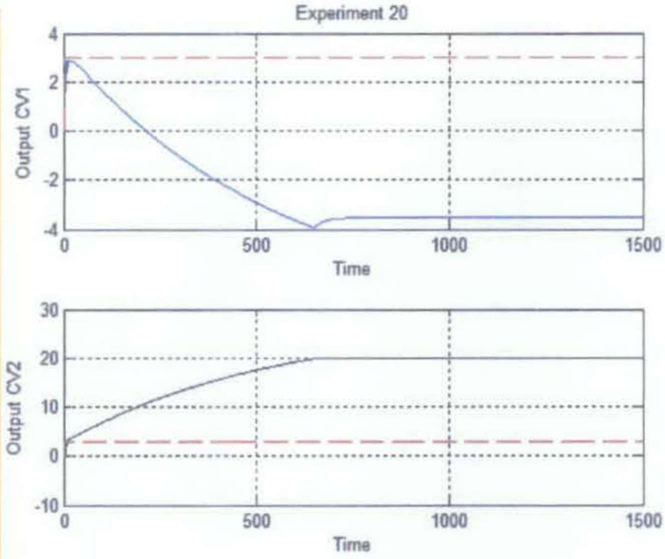

Experiment 15
Output CV1
Output CV2
Time
Time

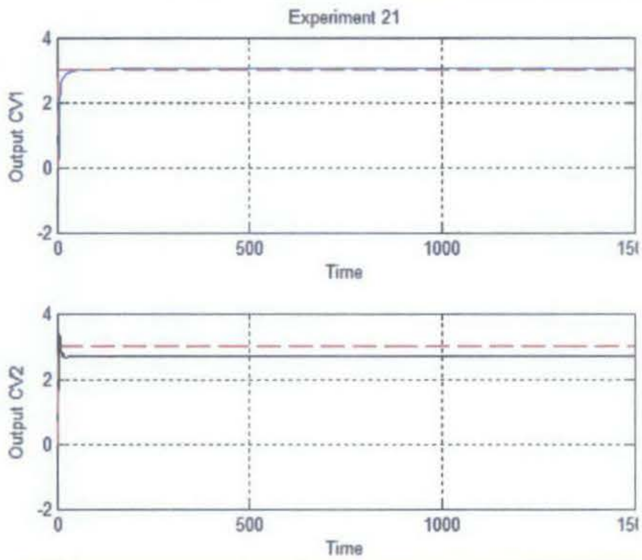

Experiment 16
Output CV1
Output CV2
Time
Time

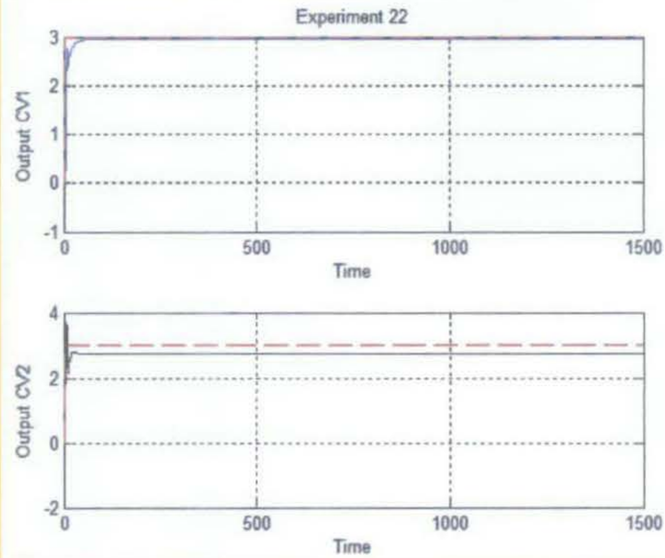

Experiment 17
Output CV1
Output CV2
Time
Time

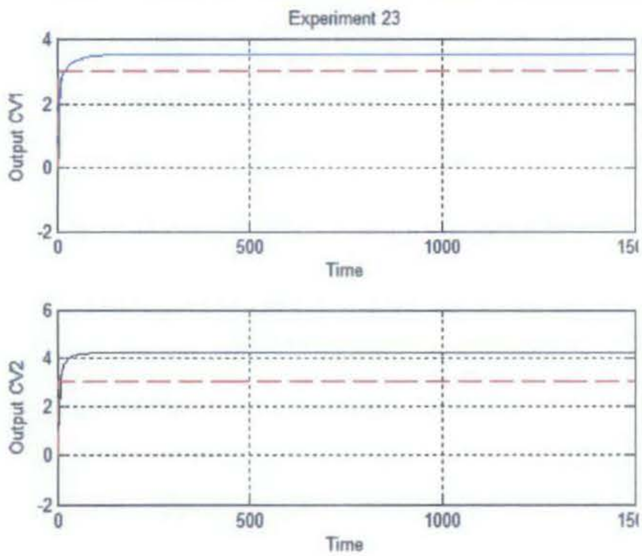

Experiment 18
Output CV1
Output CV2
Time
Time

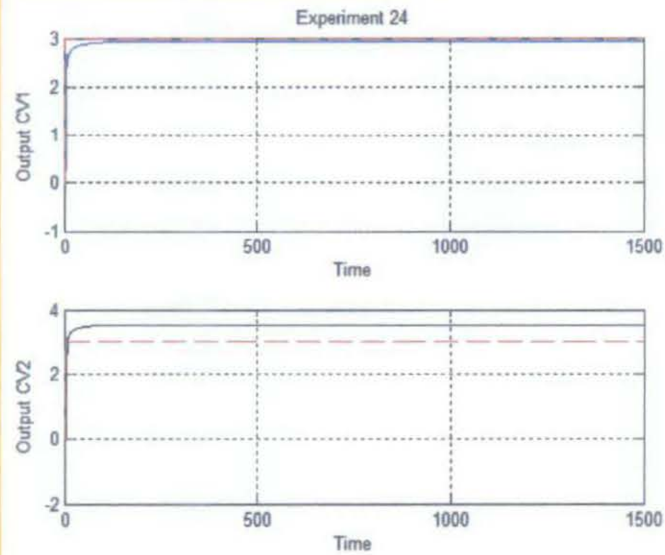

Experiment 19
Output CV1
Output CV2
Time


Experiment 20
Output CV1
Output CV2
Time


Experiment 21
Output CV1
Output CV2
Time


Experiment 22
Output CV1
Output CV2
Time


Experiment 23
Output CV1
Output CV2
Time


Experiment 24
Output CV1
Output CV2
Time

k11 values used for determining threshold limit

%	k11
0	12.8
1	12.928
2	13.056
3	13.184
4	13.312
5	13.44
6	13.568
7	13.696
8	13.824
9	13.952
10	14.08
20	15.36
30	16.64
40	17.92
50	19.2
60	20.48
70	21.76
80	23.04
90	24.32

%	k11
-1	12.672
-2	12.544
-3	12.416
-4	12.288
-5	12.16
-6	12.032
-7	11.904
-8	11.776
-9	11.648
-10	11.52
-11	11.392
-12	11.264
-13	11.136
-14	11.008
-15	10.88
-16	10.752
-17	10.624
-18	10.496

k12 values used for determining threshold limit

%	k12
-10	-17.01
-9	-17.199
-8	-17.388
-7	-17.577
-6	-17.766
-5	-17.955
-4	-18.144
-3	-18.333
-2	-18.522
-1	-18.711

%	k12
0	-18.9
1	-19.089
2	-19.278
3	-19.467
4	-19.656
5	-19.845
6	-20.034
7	-20.223
8	-20.412
9	-20.601
10	-20.79

k21 values used for determining threshold limit

%	k21
-5	6.27
-10	5.94
-20	5.28
-30	4.62
-40	3.96
-50	3.3
-60	2.64
-70	1.98
-80	1.32
-90	0.66

%	k21
0	6.6
5	6.93
10	7.26
15	7.59
20	7.92
25	8.25
30	8.58
31	8.646
32	8.712
33	8.778
34	8.844
35	8.91
36	8.976
37	9.042

k22 values used for determining threshold limit

%	k22
-1	-19.206
-2	-19.012
-3	-18.818
-4	-18.624
-5	-18.43
-6	-18.236
-7	-18.042
-8	-17.848
-9	-17.654
-10	-17.46

%	k22
0	-19.4
1	-19.594
2	-19.788
3	-19.982
4	-20.176
5	-20.37
6	-20.564
7	-20.758
8	-20.952
9	-21.146
10	-21.34

Partial correlation plot coding

```
u1=u(3:101,m);
e1=e(3:101,n);
z1=u(1:99,m);

By=(inv((z1')*(z1)))*(z1')*(u1);
ry=u1-(z1*By);
Bx=(inv((z1')*(z1)))*(z1')*(e1);
rx=e1-(z1*Bx);
c=corr(rx,ry);
crosscorr(rx,ry,[],0.5)
```